OPTIMUM DESIGN OF REINFORCED CONCRETE CHIMNEY

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

B. S. MURTY

to the

DEPARTMENT OF CIVIL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1982

CERTIFICATE

This is to certify that the thesis entitled 'Optimum Design of Reinforced Concrete Chimney' by Mr. B.S. Murty is a record work carried out under my supervision and that it has not been submitted elsewhere for a degree.

Kapoor
12/1/82

Jan. 1982

(M.P. KAPOOR)
Professor
Department of Civil Engineering
Indian Institute of Technology, Kanpur

ACKNOWLEDGEMENTS

I take this opportunity to express my deep sense of gratitude and profound regard to Dr. M.P. Kapoor for his invaluable guidance and constant encouragement which are vital to this course of work.

I want to express my thanks to my friends Messers B.V.V. Satyanarayana, Ch.S. Rao and N.V.S. Murty for their help in the final stages of the work.

Words can not express my sense of indebtedness to my beloved parents, friends and relatives, who have been a perennial source of inspiration and encouragement.

Thanks are also due to Mr. J.C. Verma for the drawing work, Mr. G S. Trivedi for his neat and elegant typing and Mr. R.S. Dwivedi for neat cyclostyling.

B.S. MURTY

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The optimum design problem of R.C. Cnimney is formulated as a multistage optimization problem. The resulting non-linear programming problem is solved by Sequential Unconstrained Minimization Technique (SUMT) using Quadratic Extended Interior Penalty Function Method. Modified Newton's method is used for unconstrained minimization.

Extended interior penalty function technique for converting constrained optimization problem into sequential unconstrained optimization problem and Modified Newton's method for solving unconstrained minimization problem are discussed in detail. The analysis and design procedures for R.C. Chimney are briefly described as, by now, these are well documented.

The efficiency of the methods used in the present work is established by comparing the results of and the C.P.U. time required to solve the same problem as given in Ref. (2, 11) where different techniques have been adopted to obtain the optimum solution.

CHAPTER 1

INTRODUCTION

1.1 Introduction:

Chimneys are relatively tall structures subjected to three types of stresses: (i) stresses due to self weight (ii) stresses due to wind or earthquake load and (iii) stresses due to temperature variation between the inside and outside of the chimney. Brick chimneys are suitable only for short heights, as they become bulky with increase in height and require heavy foundations. Also due to large temperature gradient, brick chimney frequently cracks, and becomes unstable. In contrast, concrete chimneys are lighter and stronger, and are less vulnerable to cracks due to temperature difference. If the temperature of the flue gases does not exceed 400°C, concrete chimneys can be used without any special fire brick lining. For higher temperatures, fire brick lining is provided with an air gap between the inner face of the chimney and the lining.

Most reinforced concrete chimneys have a circular cross section for the reason that a circular section affords a minimum of friction which reduces the draft. Moreover, the circular cross section is easy to construct.

The materials used in the construction of a concrete chimney are concrete, reinforcing steel and refractory bricks. Any saving in the quantity of materials used would result in the reduction of cost of chimney. An optimum design procedure for reinforced concrete circular chimneys is presented in this work with the goal to get a minimum cost design.

In general the design of reinforced concrete chimney is a trial and error process. Normally the chimney is designed by assuming mean diameter of the shell, thickness of the shell and percentage reinforcement as various sections and then the stress analysis is carried out to check if the stresses due to dead load, wind or earthquake load and temperature variation are within the permissible limits. Generally such designs are conservative involving repetitive computations and quite time consuming. Therefore, it is natural to carry out computer aided design of such structures. As already mentioned, the main forces acting at a cross-section of chimney are dead load, wind or earthquake load and temperature effect. The first three types of forces depend directly on the dimensions of the chimney above the section. Therefore, a reduction in the size of the structure would in turn reduce the forces acting on it. Thus the problem is most suitable to be formulated and solved as an optimum design problem.

The optimum design of a chimney in the present work is cast as a mathematical programming problem. The objective is to minimize the material cost of the superstructure subject to the behaviour and side constraints. The possible design variables are the mean diameter of chimney shell, thickness of the shell and reinforcement at various sections along the height of the chimney. As per code provisions six stress constraints are to be satisfied at each section. The number of design variables and the number of constraints increases directly with the number of sections to be checked and thus the size of the problem becomes large.

Chapter 2 describes the problem formulation. The optimization technique to solve the problem is described in Chapter 3. Results of a 80 m tall R.C. chimney are discussed in Chapter 4 and conclusions drawn from the present work are also summarized therein.

1.2 Previous Work:

The forces acting at a certain section of the chimney depend only on the geometry of the chimney above the section. This motivated Kapoor and Hariharan |2| to decompose the optimum design problem into a number of small sized optimization problems. In their formulation the chimney was divided into certain number of segments.

starting from the top section. At each segment a nonlinear programming problem was optimized via SUMT using interior penalty function approach and Davidon Fletcher Powell method for unconstrained minimization.

Subsequently Bandyopadhyay |11| handled the optimum design problem of R.C. chimney via dynamic programming and showed sufficient saving by way of C.P.U. time, thus establishing the superiority of the dynamic programming approach over SUMT for seeking the optimum solution of this class of problems.

1.3 Present Work:

The present work addresses itself

(i) to solve the multistage optimization problem of R.C. chimney by converting the resulting constrained nonlinear programming problem into unconstrained minimization problem through extended interior penalty function approach and solve the resulting unconstrained problem by Modified Newton's Method ;

(ii) to compare the optimum design so obtained with the optimum solution reported in Ref. |2, 11| and

(iii) to compare the relative efficiency of the optimization techniques used herein with the other two techniques used in Ref. |2,11| for this class of structure.

The cost of concrete and steel are only considered in the objective function as in the earlier two works cited. Furthermore, the Indian Standards Specifications |1,3| have been followed in formulating the constraints so as to compare the results of the present work with those reported earlier.

CHAPTER 2

PROBLEM FORMULATION

2.1 Formulation of the Problem:

The optimum design of a chimney, in the present work, consists in minimizing the material cost of its superstructure subject to the constraints imposed by the code provisions |1,3|. The resulting mathematical programming problem is to find the design vector $\vec{X}$ which minimizes the objective function

$$C = f(\vec{X})$$

subject to behaviour constraints

$$g_j(\vec{X}) \geq 0, \quad j = 1, \ldots\ldots, m \qquad (2.1)$$

and side constraints

$$x_i^l \leq x_i \leq x_i^u, \quad i = 1, \ldots\ldots, n$$

The objective function is taken as the cost of concrete shell and vertical reinforcement used in the superstructure. The mean diameter, thickness of concrete shell and percentage of vertical reinforcement are taken as the three continuously varying design variables. In terms of the variables, the cost of concrete shell can be expressed as

$$C = \int_0^H C_u \, \pi \, d(h) \, t(h) \, (1 + Sp(h)) \, dh \qquad (2.2)$$

where $d(h)$, $t(h)$ and $p(h)$ are the mean diameter, the shell thickness and percentage of reinforcement at a section of distance h from top of the chimney. H is the total height of the superstructure, C_u is the cost of unit volume of concrete and S is the ratio of the cost of unit volume of steel to that of concrete.

By converting the integral in Eq. (2.2) to a summation form, we get

$$C_1 = C_u \sum_{i=1}^{N+1} \pi\, d_i\, t_i (1 + S\, p_i)\, h_i \qquad (2.3)$$

where the index i denotes the ith section of the chimney which is divided along its height into N equal parts. It is assumed that the mean diameter and shell thickness of the chimney vary linearly between two consecutive sections.

Eq. (2.3) can be rewritten as

$$C_1 = C_u \sum_{i=1}^{N+1} A_i\, h_i \qquad (2.4)$$

where $A_i = \pi\, d_i\, t_i (1 + S\, p_i)$

and h_i is the distance between the ith and (i-1)th sections (Fig. 1). For i=1 i.e. the top section, the mean diameter is fixed from nonstructural design considerations. The thickness of the shell, t_1, and percentage of steel, p_1, at the top section have been

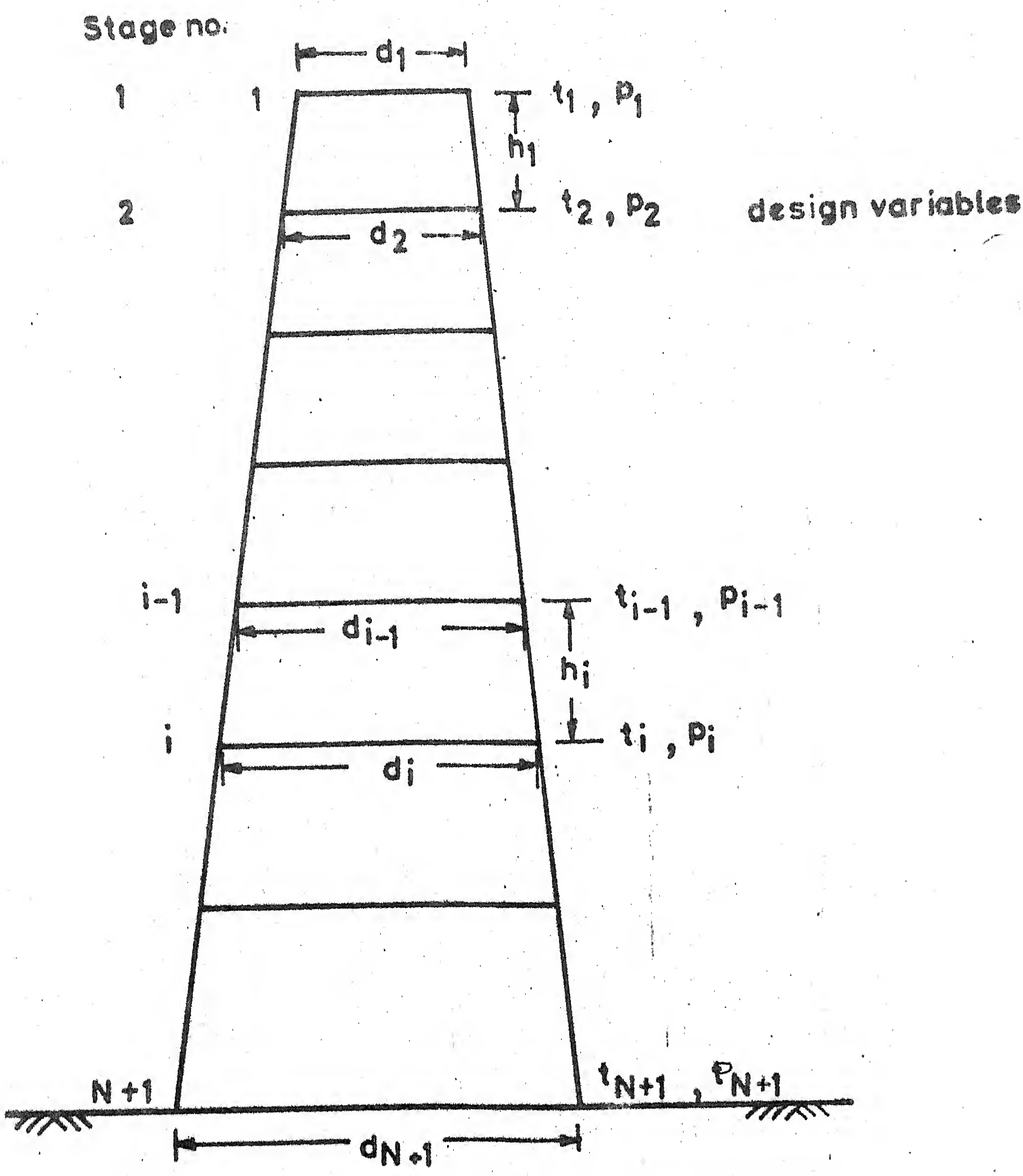


FIG.1 THE CHIMNEY DESIGN PROBLEM DECOMPOSED TO A MULTISTAGE DESIGN PROBLEM

set to the minimum. The resulting optimization problem would have 3N design variables, viz., d_i, t_i and p_i, $i=2, \ldots, N+1$. As the size of the problem turns out to be large a kind of decomposition procedure was adopted in Ref. |2| and the same is being followed in the present work. The dead load and wind load at any ith section depend evidently upon the dimensions of the chimney above that section. Hence it can be argued that if one proceeds to optimize the equivalent area of cross section, A_i, sequentially, starting from the top, the resulting design, though not the theoretical optimum design of the entire structure, still approaches it from the upper side.

$$\text{Min } C_1 = \text{Min } (C_u \sum_{i=1}^{N+1} A_i h_i) \leq C_u \sum_{i=1}^{N+1} h_i (\text{Min } A_i) \qquad (2.5)$$

This formulation consists of sequentially minimizing $A_i = 2, \ldots, N+1$, subject to the behaviour constraints on the chimney and side constraints on the design variables. For each section, the objective function is the equivalent area of cross section of concrete

$$A_i = \pi \, d_i \, t_i \, (1 + S \, p_i) \qquad (2.6)$$

A_i represents the equivalent area of concrete used in the computation of the cost of the superstructure.

The problem has thus been decomposed to N problems each with three design variables d_i, t_i and p_i. The

behaviour constraints are the bounds on stresses in each of the decomposed problem and the side constraints are the bounds on design variables as specified in Indian Standard Specifications |1|.

2.2 Code Specifications:

2.2.1 Loading conditions:

The Indian Standard Specification for the design of R.C. Chimney |1| requires that the chimney section at all heights should be safe against the following load conditions.

(i) Dead load + wind load

(ii) Dead load + earthquake force

(iii) Dead load + temperature effect

(iv) Dead load + wind load + temperature effect

(v) Dead load + earthquake force + temperature effect.

2.2.2 Permissible stresses:

The permissible stresses in concrete and steel for each of the above loading cases are specified in the code |1| and are given in Table 1 for the sake of completeness of presentation.

TABLE 1

PERMISSIBLE STRESSES

Load condition		Permissible stresses (kg/cm^2)	
		Concrete	Steel
(i)	Dead load +wind load	$0.38\ f_c$	$0.57\ f_{sy}$
(ii)	Dead load + earthquake force	$0.40\ f_c$	$0.60\ f_{sy}$
(iii)	Dead load + temperature effect	$0.33\ f_c$	$0.55\ f_{sy}$
(iv)	Dead load + wind load + temperature effect	$0.50\ f_c$	$0.65\ f_{sy}$
(v)	Dead load + earthquake force + temperature effect	$0.50\ f_c$	$0.65\ f_{sy}$

where f_c is the 28- days cube strength of concrete in kg/cm^2

f_{sy} is the yield stress of steel in kg/cm^2.

2.2.3 Minimum requirements:

(i) Minimum thickness of the shell $(t_i) = 15$ cm

for diameter $(d_i) \leq 6$ m

$$t_i \geq 15 + \frac{d_i - 6}{1.2} \text{ cm if } d_i > 6 \text{ m}$$

(ii) Percentage of reinforcement

$$0.003 \leq p_i \leq 0.05$$

(iii) Minimum clear cover = 5 cm.

2.2.4 Stress calculations:

At any height of the chimney the dead load, wind and earthquake load can be calculated once the geometry of the chimney above the section is known and the stress in each case can be computed as discussed in the succeeding paragraphs. Among the wind load and earthquake load, only the more severe of the two at a cross-section* is considered for the purposes of stress analysis.

---

* It was observed by Kapoor and Hariharan |2|, that the wind loads are in general predominant in the lower half of the chimney while the earthquake loads dominate in the upper half. The nature of wind and earthquake moments along the height of the chimney considered in the present work is shown in Fig.(2) which is taken from Ref. (2).

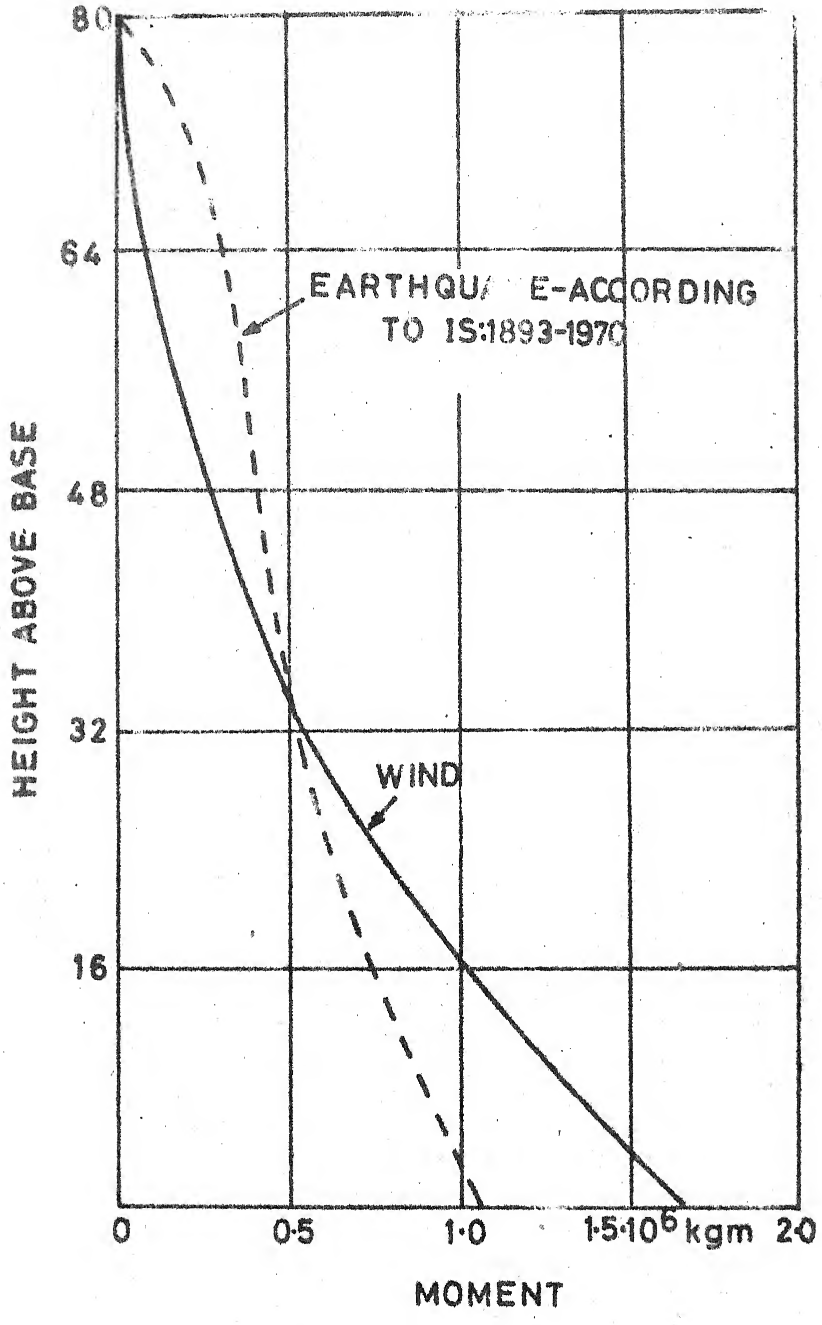


FIG. 2 NATURE OF WIND AND EARTHQUAKE MOMENTS

(i) Stresses due to dead load only

The maximum vertical stress in concrete due to dead load only is given by

(a) For annular section with no opening

$$f'_{cvd} = \frac{W}{2\pi rt} \tag{2.7}$$

(b) For annular section with one opening (Fig.3)

$$f'_{cvd} = \frac{W}{2(\pi-\beta)tr} \tag{2.8}$$

The maximum vertical tensile stress in steel is given by,

$$f_{svd} = m\, f'_{cvd} \tag{2.9}$$

(ii) Vertical stresses due to temperature only

The difference in temperature between the two faces of chimney shell is given by

$$T_x = \frac{t\, D_{bi}}{C_c\, D_c} \left[\frac{T - T_o}{\dfrac{1}{r_q k_1} + \dfrac{t_b\, D_{bi}}{r_q\, C_b\, D_b} + \dfrac{D_{bi}}{k_s\, D_s} + \dfrac{tD_{bi}}{C_c D_c} + \dfrac{D_{bi}}{k_2 D_{co}}} \right] \tag{2.10}$$

where T is the maximum temperature of gas inside chimney

T_o is the minimum temperature of outside air surrounding the chimney.

r_q, C_c, C_b, k_1, k_2 and k_s are coefficients depending

on thermal properties of concrete, lining and insulating medium (air). The geometric parameters appearing in Eq. (2.10) are shown in Fig. (3).

The vertical stress in steel, f_{STV}, and in concrete, f_{CTV}, due to the temperature gradient through the shell (Fig. 5), are given by

$$f_{STV} = \alpha \; (Z - K) \; T_x \; E_s \qquad (2.11)$$

$$f_{CTV} = \alpha \; K \; T_x \; E_c \qquad (2.12)$$

and

$$K = - pm + \sqrt{pm(pm+2Z)} \qquad (2.13)$$

where,

- α is the coefficient of thermal expansion of concrete and steel per degree centigrade.
- Z is the ratio of distance between inner surface of chimney shell and vertical reinforcement to total shell thickness t.
- E_s is modulus of elasticity of steel, and
- E_c is modulus of elasticity of concrete
- m is the modular ratio
- p is the ratio of total area of vertical reinforcement to total area of concrete chimney shell at the section under consideration.

(iii) <u>Moment due to earthquake:</u>

The procedure for calculating moments due to earthquake forces for stacklike structures is laid down in section 5.3

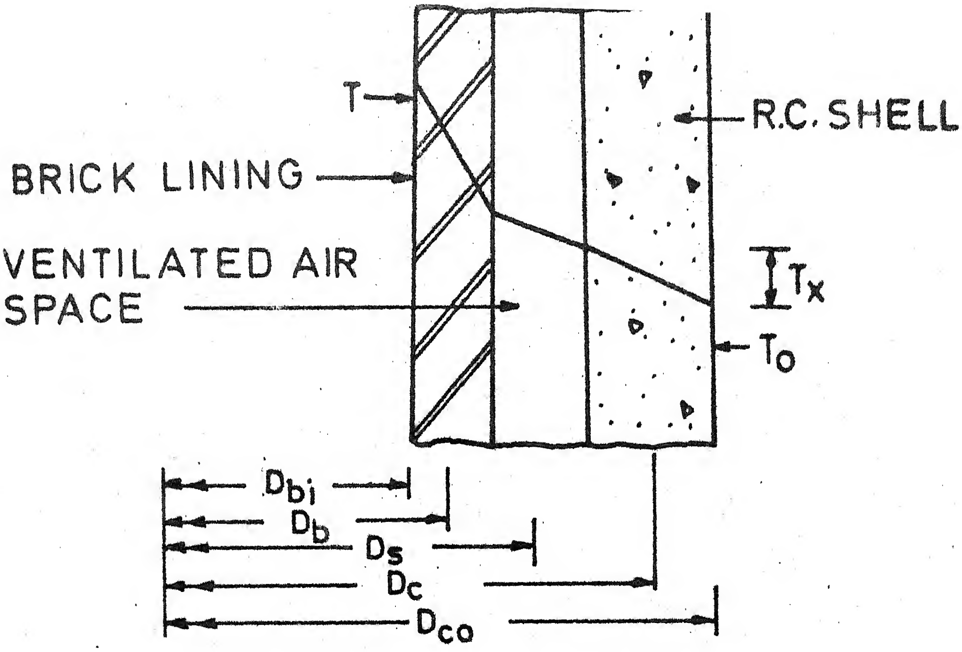


FIG. 3 TEMPERATURE GRADIENT THROUGH SHELL THICKNESS

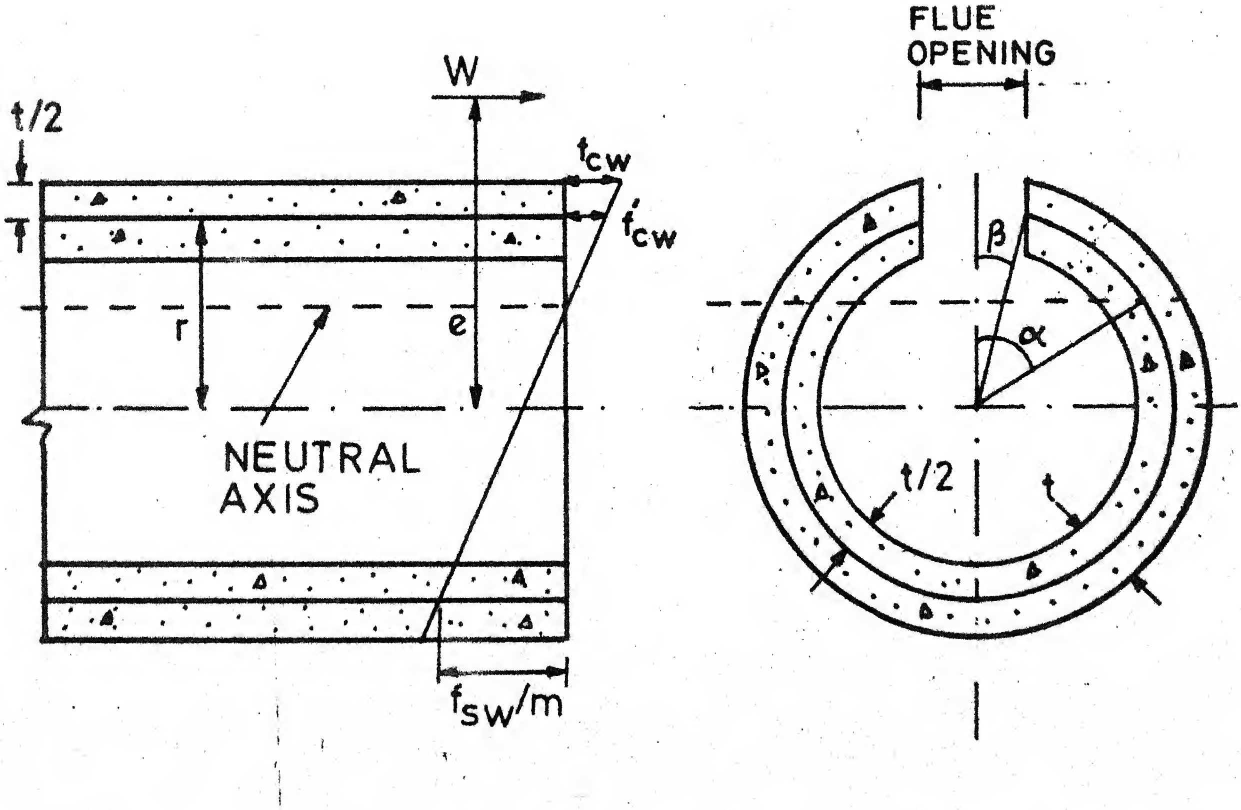


FIG. 4 VERTICAL STRESS DUE TO LATERAL LOAD & DEAD LOAD

of earthquake code |3| .

The approximate fundamental period of free vibration, T in sec. is calculated from

$$T = C_T \sqrt{\frac{W \quad H}{E \, A \cdot g}} \qquad (2.14)$$

where,

C_T = A coefficient based on slenderness ratio of chimney given in Ref. |3|.

W = Total weight of the chimney including weight of lining above the base.

H = Height of chimney above the base

E = Modulus of elasticity of material of the structural shell

A = Area of cross-section of structural shell of the base

g = Acceleration due to gravity.

Using period T, the horizontal seismic coefficient α_h is obtained from

$$\alpha_h = \beta I F_o \frac{S_a}{g} \qquad (2.15)$$

where, I = Importance factor for the structure

β = A coefficient depending on soil foundation system

F_o = Basic horizontal seismic coefficient given in Ref. |3|

$\frac{S_a}{g}$ = **Average** acceleration coefficient from average acceleration spectra given in code and reproduced in Fig. (5).

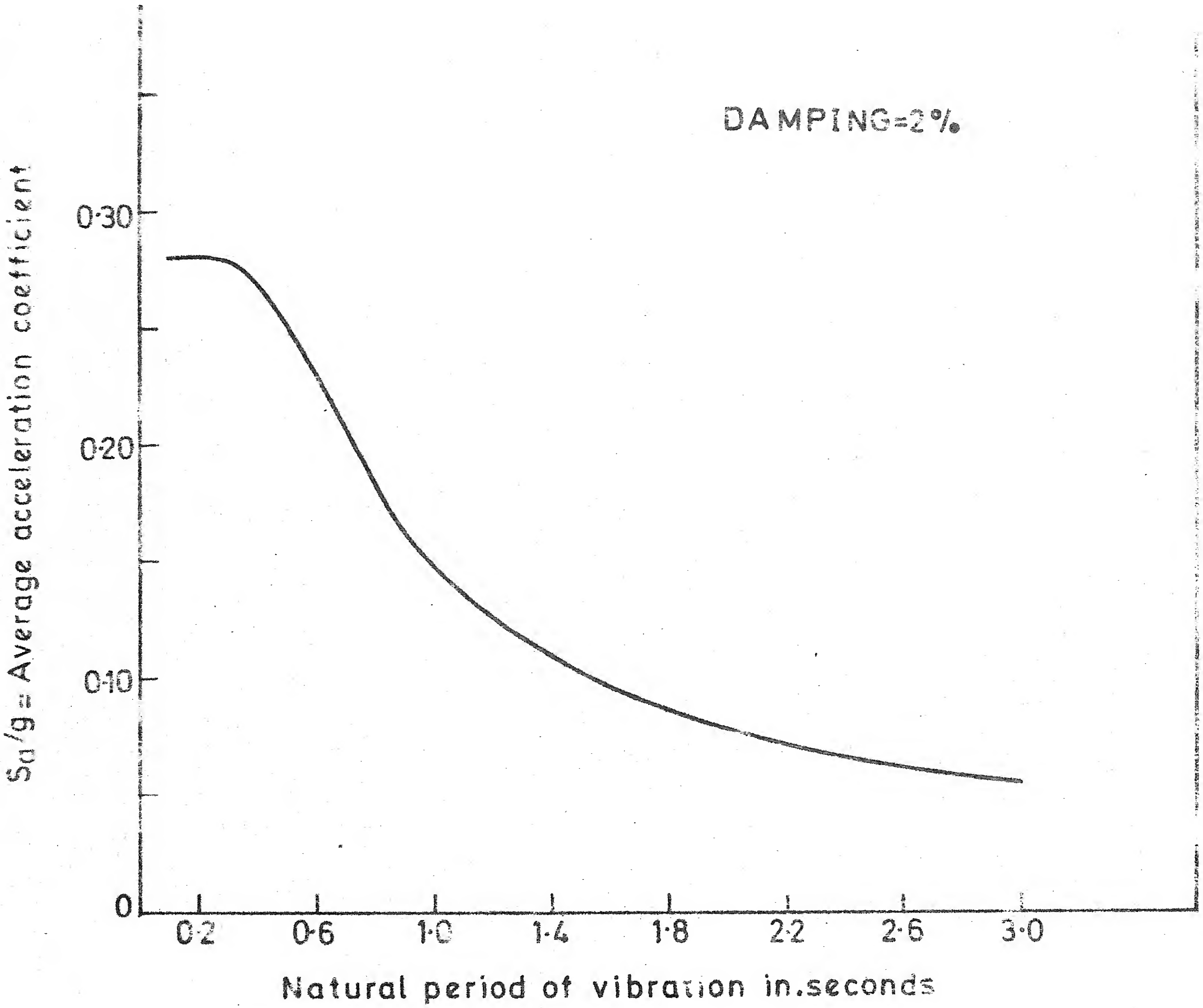


FIG. 5 AVERAGE ACCELERATION SPECTRA

Then the design bending moment M_x at a distance x from top is calculated by the following formula

$$M_x = \alpha_h W \bar{h} \left[0.6(x/H)^{1/2} + 0.4(x/H)^4\right] \qquad (2.16)$$

where

$\bar{h}$ is the centre of gravity of structure above the base.

(iv) Stresses due to lateral load and dead load:

The whole section of shell is under compression, if

(a) For annular sections

$$\text{if} \quad e/r \leq 1/2 \qquad (2.17)$$

The maximum vertical compressive stress in concrete shell is given by

$$f_{cw} = \frac{W}{2\pi r t}\left[1 + \frac{2e}{r}\right] \qquad (2.18)$$

where,

e is the eccentricity

r is the mean radius of the shell at the section under consideration

t is the thickness of the shell at the section under consideration

W is the weight of the chimney above the section.

(b) For annular section with one opening (Fig.4)

$$\frac{e}{r} \leq \frac{1}{2(\pi-\beta)}\left[\frac{(\pi-\beta)^2 - \sin^2\beta}{(\pi-\beta)\cos\beta + \sin\beta} - 3\sin\beta\right] \qquad (2.19)$$

Then in such case, the maximum vertical compressive stress is concrete is given by,

$$f_{cw} = \frac{W}{2(\pi-\beta)rt}\left[1 + \frac{\left\{2\frac{e}{r} + \frac{\sin\beta}{(\pi-\beta)}\right\}\left\{(\pi-\beta)\cos\beta+\sin\beta\right\}}{(\pi-\beta) - 0.5\sin 2\beta - 2\sin^2\beta/(\pi-\beta)}\right] \tag{2.20}$$

If e/r is greater than the corresponding right hand side of expressions (2.17) or (2.19), the angle α(Fig.4) defining the position of the neutral axis of the R.C. shell section is calculated by numerically solving the transcendental equation given below

$$\frac{e}{r} = \frac{A}{2B} \tag{2.21}$$

where,

$$A = (1-p)(\alpha-\sin\alpha\cos\alpha)-(1-p+mp)(\beta+\sin\beta\cos\beta - 2\cos\alpha\sin\beta) + mp\pi$$

$$B = (1-p)(\sin\alpha-\alpha\cos\alpha)-(1-p+mp)(\sin\beta-\beta\cos\alpha)-mp\pi\cos\alpha$$

m is the modular ratio

β is angle in radians of semi-flue opening

p is ratio of total area of vertical reinforcement to total area of concrete chimney shell at the section under consideration.

Regula falsi method has been used to find the value of α. Once α is known, the compressive stress in concrete

at mid fibre of the section (Fig.4), f'_{cw}, is calculated as,

$$f'_{cw} = \frac{W}{2rt}\left[\frac{(\cos\beta - \cos\alpha)}{(1-p)(\sin\alpha - \alpha\cos\alpha) - (1-p+mp)(\sin\beta - \beta\cos\alpha) - mp\pi\cos\alpha}\right] \quad (2.22)$$

The maximum compressive stress in concrete, f_{cw}, becomes,

$$f_{cw} = f'_{cw}\left[1 + \frac{t}{2r\ \cos\beta(\cos\beta - \cos\alpha)}\right] \quad (2.23)$$

The maximum tensile stress in steel, f_{sw}, is given by,

$$f_{sw} = mf'_{cw}\left[\frac{1 + \cos\alpha}{\cos\beta - \cos\alpha}\right] \quad (2.24)$$

(v) Stress due to dead load and temperature

In the case of reinforced concrete, the stresses cannot generally be added arithmetically but must be combined according to the elastic properties of the materials and in the proportion of reinforcement to concrete.

The maximum vertical compressive stress in concrete due to combined effect of vertical load plus temperature effect is given by,

$$f_{cvdt} = f_{CIV}\frac{K_{dt}}{K} \quad \text{if } K_{dt} \leq 1 \quad (2.25)$$

or

$$f_{cvdt} = f'_{cvd} + \frac{f_{CTV}}{K} \left[\frac{2mpZ + 1}{2\,(1+mp)} \right] \text{ if } K_{dt} \geq 1 \qquad (2.26)$$

where,

$$k_{dt} = -mp + \sqrt{mp(mp+2Z)+2K(1+mp)\frac{f'_{cvd}}{f_{CTV}}}$$

k_{dt} is the ratio of distance between inner surface of the chimney shell and the neutral surface resulting from combined dead load and temperature, to the total shell thickness, .t.

The maximum vertical tensile stress in steel due to combined effect of dead load and temperature is given by,

$$f_{svdt} = \frac{f_{STV}}{(Z-K)} \left[Z+mp \;-(mp(mp+2Z)-2mp(Z-K)\frac{f_{svd}}{f_{STV}})^{1/2} \right] \qquad (2.27)$$

(vi) <u>Stresses due to dead load, lateral load and temperature</u>:

The stresses in the concrete and reinforcement can be computed for each case, i.e. on windward side and leeward side of the chimney. It is clear that the design will be governed by the worst cases, namely (1) compression in concrete on the leeward side of the shell and (2) tension in the reinforcement on the windward side of the shell.

The maximum compressive stress in concrete due to combined effect of lateral load, dead load and temperature Figs. (6 and 7) is given by

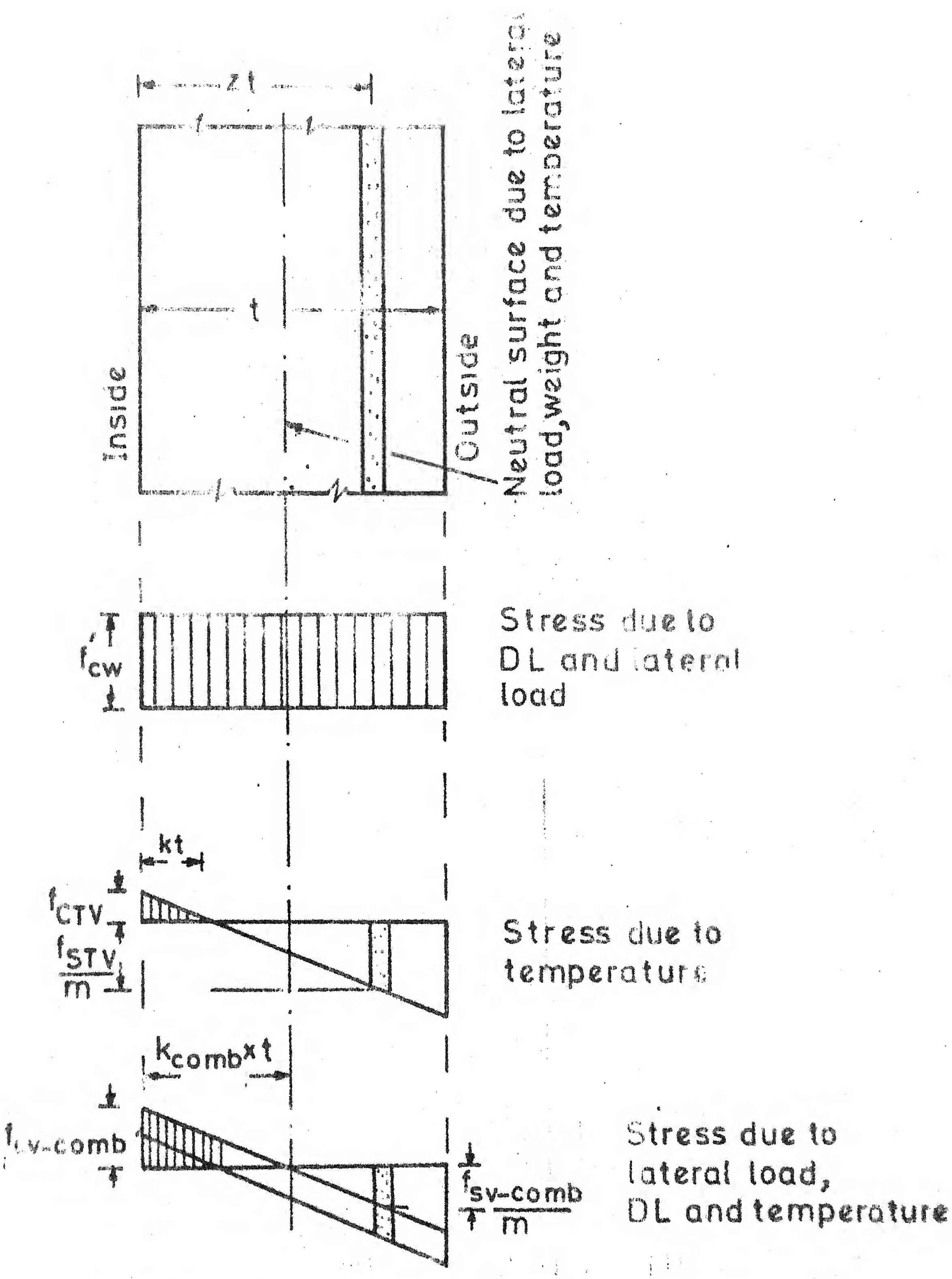


FIG. 6 STRESS DISTRIBUTION ON THE LEEWARD SIDE OF THE CHIMNEY ($K_{comb} \leq 1$)

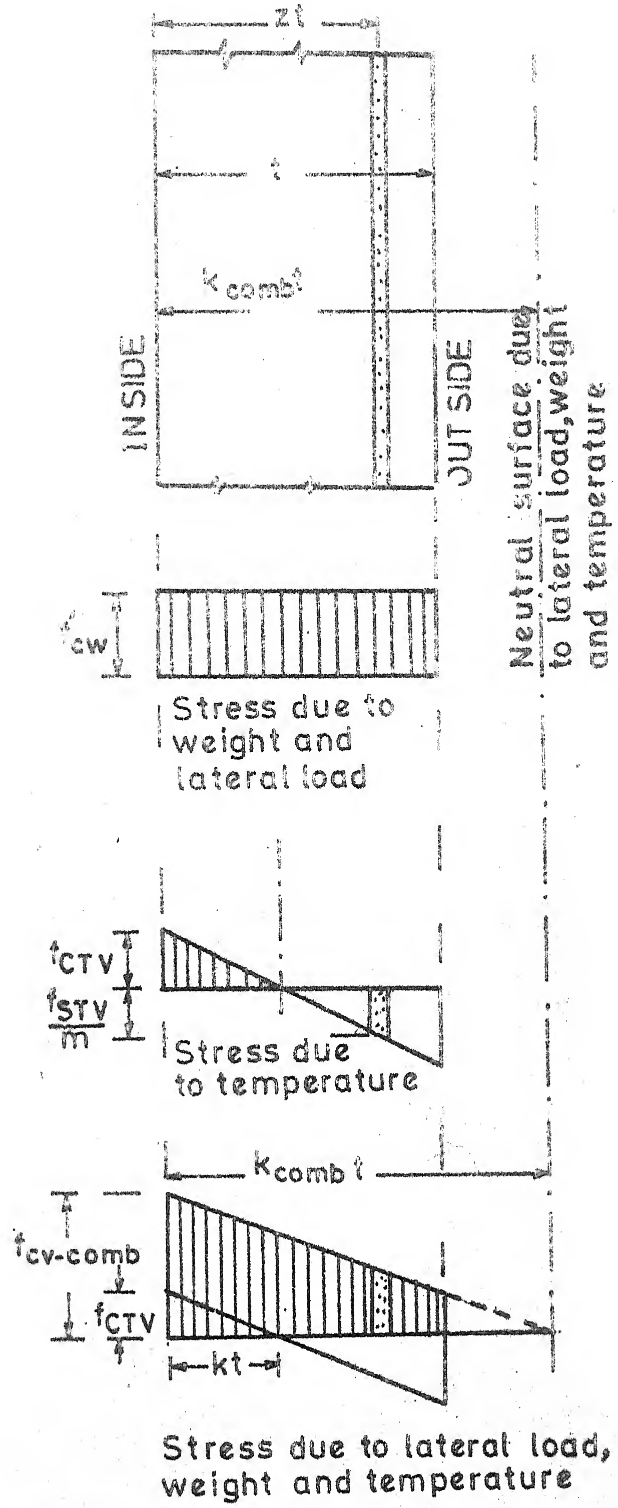


FIG. 7 STRESS DISTRIBUTION ON THE LEEWARD SIDE OF THE CHIMNEY ($k_{comb} \geq 1$)

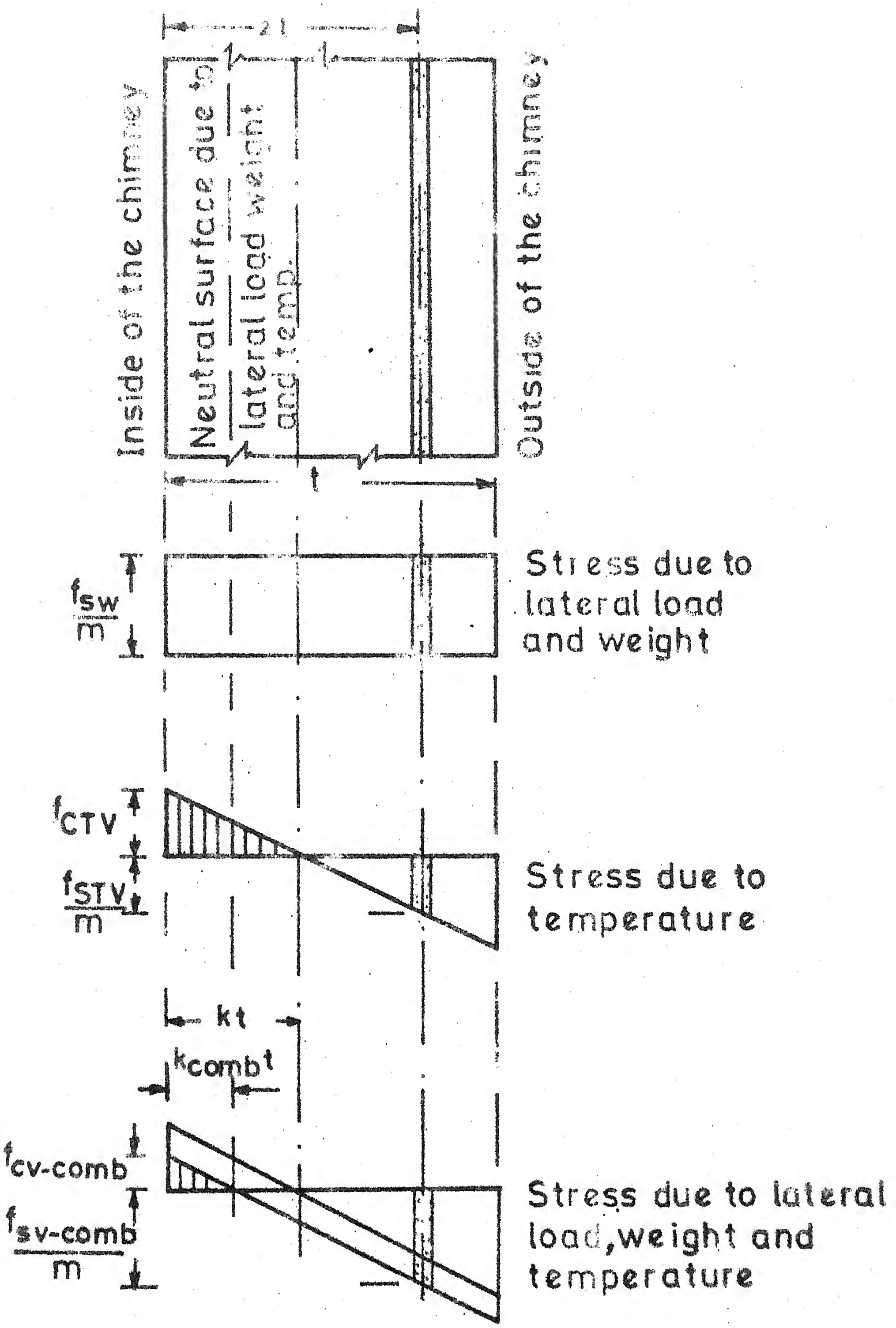


FIG.8 STRESS DISTRIBUTION ON THE WINDWARD SIDE OF THE CHIMNEY

$$f_{cw-comb} = \frac{f_{CTV} K_{comb}}{K} \quad \text{if } K_{comb} \leq 1 \qquad (2.28)$$

or

$$f_{cw-comb} = f'_{cw} + \frac{f_{CTV}}{K} \left[\frac{2mpZ + 1}{2 (1 + mp)} \right] \quad \text{if } K_{comb} \geq 1 \qquad (2.29)$$

Where K_{comb} is the ratio of distance between inner surface of chimney shell and neutral surface resulting from combined lateral load, dead load and temperature effect to shell thickness, and is calculated as

$$K_{comb} = -pm + \left| pm (pm+2Z) + 2K (1+pm) \frac{f'_{cw}}{f_{CTV}} \right|^{1/2} \qquad (2.30)$$

The maximum tensile stress in reinforcing steel, $f_{sw-comb}$, in the windward side of the chimney, (Fig. 8), is given by

$$f_{sw-comb} = \frac{f_{STV}}{(Z-K)} \left[(Z+pm) - \left(pm(pm+2Z)-2pm(Z-K) \frac{f_{sw}}{f_{STV}} \right)^{1/2} \right] \qquad (2.31)$$

2.3 <u>Compatibility Requirement:</u>

For the purposes of a compatible and practical structure, the mean diameter plus the thickness of the shell at ith section should be greater than or equal to the mean diameter and the thickness of the shell at (i-1)th section i.e.,

$$d_i + t_i \geq d_{i-1} + t_{i-1} \qquad (2.32)$$

2.4 Optimum Design Problem:

The minimum cost design of R.C. chimney has thus been formulated as a multistage optimum design problem. It consists of finding the solution of N optimum design problems each consisting of 3 design variables, 6 behaviour constraints and 4 side constraints as follows:

$$\text{minimize} \quad A_i(\vec{X}) = \pi\, d_i\, t_i\, (1 + S\, p_i)$$

$$i = 2, \ldots\ldots N + 1$$

$$\text{where} \quad \vec{X}^T \equiv (d, t, p)$$

Subject to behaviour constraints, given by the limitations on stress defined in sections 2.2.2 and 2.2.4 and side constraints defined in sections 2.2.3 and 2.3.

2.5 Normalization of Constraints:

All the constraint equations are normalized such that they are non-dimensional and lie in the range of 0 to 1. Typical examples of normalizing a behaviour and side constraints are shown below.

A behaviour constraint is typically represented as

$$S \leq S_{max}$$

Rewriting the above inequality

$$S_{max} - S \geq 0$$

Dividing both sides by S_{max}, the normalized constraint can be obtained as

$$g_j = 1 - \frac{S}{S_{max}} \geq 0$$

A similar procedure is adopted for normalizing the side constraints also. In the present formulation all the constraints (g_js) are expressed in greater than or equal to zero form.

CHAPTER 3

OPTIMIZATION TECHNIQUE

3.1 Introduction:

In the interior penalty function formulation for solving inequality constrained minimization problems, the constrained minimization problem is transformed into a sequence of unconstrained minimization problems. It is reliable and it facilitates the use of efficient unconstrained minimization algorithms to solve the inequality constrained problem. Moreover, in structural design optimization the method exhibits a desirable property, from a practical viewpoint, of generating a sequence of feasible designs that tend to funnel down to the middle of the feasible region. As a result of this property each design generated defines a structure for which the excess weight, with respect to the minimum, is distributed so that none of the constraints become critical prior to converging to an optimum design.

The interior penalty function is only defined within the feasible domain. It is known that if the starting point is feasible, then the method should generate only feasible points in the sequence of designs converging toward the optimum. However, most of the powerful algorithms for

unconstrained minimization involve a sequence of one-dimensional minimizations and in practice they are carried out by the numerical methods which may well generate trial design points outside of the feasible region. Extending the penalty function formulation into the infeasible design space provides an efficient mechanism to recover from violations brought about by analysis and also permit the use of an infeasible initial design as a starting point. Numerous recovery schemes that can bring a design from infeasible design space back to feasible design space are available. But it is believed, Ref. |14| , that these recovery schemes are not as efficient as the extended penalty function approach.

3.2 Motivation for Extended Interior Penalty Function Method:

Caroll proposed the interior penalty function method to convert the constrained non-linear optimization problem into unconstrained problem. In general the constrained optimization problem may be posed as

Minimize $F(\vec{X})$

Subject to constraints $g_j(\vec{X}) \geq 0$, $j=1,\ldots n_{con}$.

In the interior penalty function formulation a new function. *The penalty term is chosen such that its value

$P(\vec{X}, r)$ is constructed by augmenting a penalty term to the objective function.

will be small at points away from the constraint boundaries and will tend to infinity as the constraint boundaries are approached. The penalty function P ($\vec{X}$, r) is given by Caroll [19]

$$P(\vec{X}, r) = F(\vec{X}) + r \sum_{j=1}^{N_{con}} G_j(\vec{X}) \quad (3.2)$$

$$G_j(\vec{X}) = \frac{1}{g_j(\vec{X})}$$

$$\text{for } g_j(\vec{X}) \geq 0$$

The term r $G_j(\vec{X})$ represents the penalty associated with the jth constrain, and is an interior penalty function in the sense that it is defined only if $\vec{X}$ is inside the feasible design domain. With $\vec{X}$ denoting the point in the design space, P ($\vec{X}$, r) attains its minimum value for a given value of r. It is shown (Ref.8) that as r goes to zero

$$\text{min. } P(\vec{X},r) \rightarrow F(\vec{X}^*)$$

$$\text{and } \quad \vec{X} \rightarrow \vec{X}^* .$$

Therefore, the penalty function is sequentially minimized for a decreasing sequence of r until r approaches zero. This is called sequential unconstrained minimization technique (SUMT) and uses any of the known unconstrained minimizers.

The variation of the ordinary interior penalty function along a direction $\vec{S}$ in the design variable space is shown in Fig. 9, in which α represents the scalar variable that denotes the distance along $\vec{S}$. It is assumed that the minimum point M in the direction $\vec{S}$, is to be found by a numerical method, say, cubic interpolation. It will therefore, be necessary to locate two trial designs one on either side of the minimum M'. Steps are to be taken in the direction of increasing α to find a point on the portion MA of the curve. This search becomes an increasingly difficult problem because the interval M'N' tends to become very narrow as the SUMT algorithm advances toward the optimum design. Therefore, it is likely that the search for α values in the interval M' N' will produce trial design points that are outside of the feasible region. Techniques for stepping back into the region M' N' usually require a large number of function evaluation. Every function evaluation entails structural analysis and thus the procedure of stepping back introduces a heavy computational burden. The troubles do not end once a point in the M'N' region is found. This is because the function value and the first derivative of $P(\vec{X},r)$ with respect to α will frequently be very large at the located point in the interval M'N'. These high values are not well suited to construct a good cubic approximation of $P(\vec{X},r)$. Therefore, the minimum of the cubic approximation

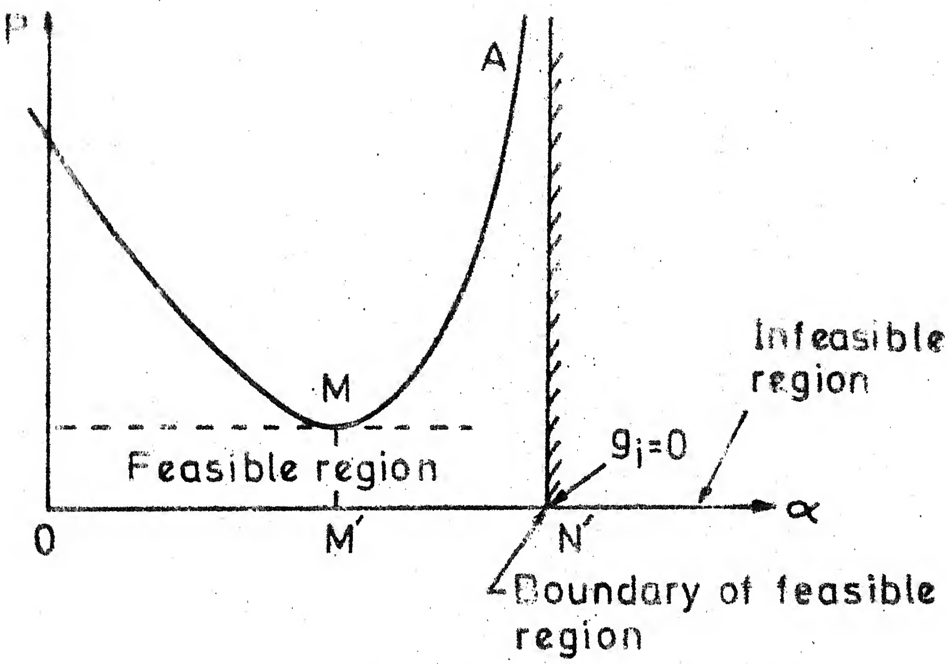


FIG.9 ONE DIMENSIONAL MINIMIZATION OF THE INTERIOR PENALTY FUNCTION

does not yield a good estimate of the α value corresponding to the actual solution of the one-dimensional minimization problem.

These important drawbacks motivated Casis and Schmit | Ref. 13| , the idea of devising an extended penalty function which is defined in the infeasible region, while still retaining the advantages of an interior penalty function formulation. The extended penalty function formulation accomodates the infeasible design points, avoiding the costly search for a feasible design in the narrow region M' N'.

3.3 Linear Extended Interior Penalty Function:

Kavlie and Moe [20] proposed a formula for the extended penalty function. The transition point in the interior penalty function satisfies continuity of both function value and its first derivative. They define the individual constraint contribution to the penalty term of the compound interior-extended penalty function as follows

$$G_j(\vec{X}) = \frac{1}{g_j(\vec{X})} \quad \text{for } g_j(\vec{X}) \geq g_o$$

$$= \frac{2g_o - g_j(\vec{X})}{g_o^2} \quad \text{for } g_j(\vec{X}) \leq g_o \qquad (3.3)$$

where g_o is a transition point between the two constraint functions and varies as

$$g_o \quad = \quad C\, r \tag{3.4}$$

where C is a constant and r is the penalty parameter.

Because the G_j of Eq. (3.3) have discontinuous second derivatives at the transition point, the linear extension of the penalty function is not suitable for a second order (e.g. Newton's method) optimization algorithms.

The formula proposed by Kavlie and Moe (Eq. (3.3)) defines the extended penalty part of each constraint g_J. The key parameter g_o corresponds to the values of the g_j's at which the transition between the regular interior penalty function definition and the extended penalty function occurs. From the one-dimensional view of the interior penalty function shown in Fig. (9), it can be seen that definition of a useful extended penalty function requires the transition point to lie on the MA portion of the curve, the reason for this requirement is that the extended interior penalty function would not include the minimum at point M and indeed its minimum (M_E) might even be outside of the feasible region, as shown in Fig. (10). Now, recalling that the introduction of the extended penalty function was motivated to locate the minimum along linear direction, it is obvious

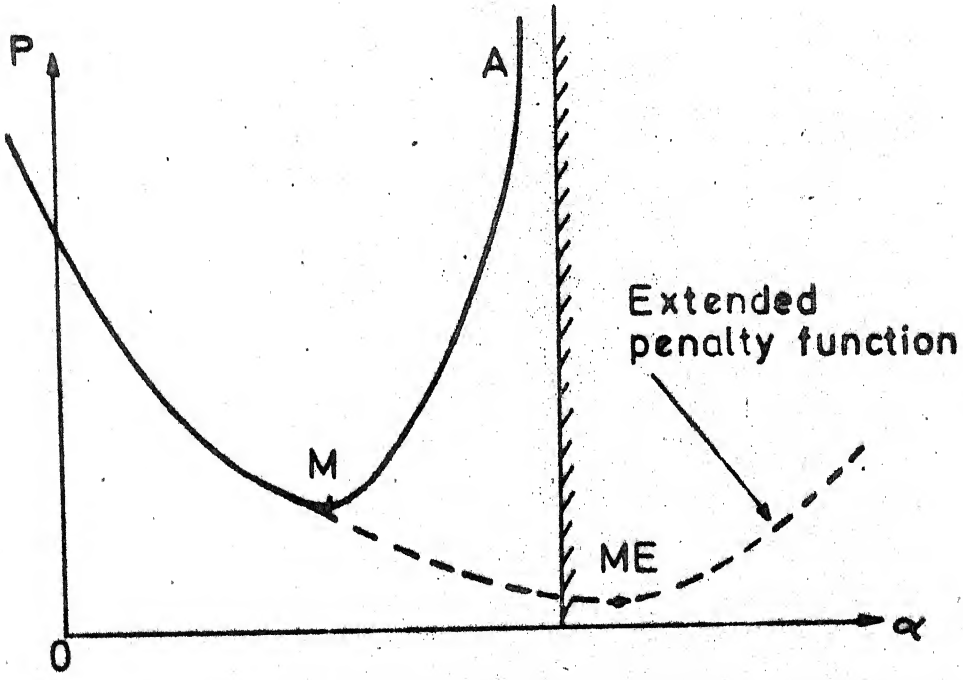


FIG.10 EXTENDED PENALTY FUNCTION ON THE WRONG SIDE OF THE MINIMUM

that extended penalty function should be defined so as to include the minimum of original penalty function. A more rational approach to find g_o is described in (Ref.13). However, since it is not pertinent to the present work, it is not being discussed here.

3.4 Quadratic Extended Interior Penalty Function:

The disadvantage of the discontinuous second derivatives at the transition point in the linear extended penalty function can be overcome by introducing a quadratic extended interior penalty function (Ref. 14). It is continuous and has continuous first and second derivatives. The definition of $G_j(\vec{X})$ in Eq. (3.2) for a quadratic extended penalty function is given by,

$$G_j = \frac{1}{g_j(\vec{X})} \quad \text{if } g_j(\vec{X}) \geq g_o$$
$$= \frac{1}{g_o}\left[\left(\frac{g_j(\vec{X})}{g_o} - 1\right)^2 - \left(\frac{g_j(\vec{X})}{g_o} - 1\right) + 1\right] \quad \text{if } g_j(\vec{x}) \leq g_o \tag{3.5}$$

The transition point g_o for the above quadratic extension to the penalty function is assumed (Ref. 14) as

$$g_o = C\, r^p \; ; \quad 1/3 < p \leq 1/2 \tag{3.6}$$

where C is a constant.

As r goes to zero, the above equation has to satisfy the conditions:

(i) $r\ G_j(\vec{X}) \rightarrow 0$ for any $g_j(\vec{X}) > 0$,

which represents a vanishing contribution in the feasible domain and

(ii) $r\ G_j(\vec{X}) \rightarrow \infty$ for any $g_j(\vec{X}) < 0$,

which represents an increasing penalty for violated constraints.

The lowerbound on the exponent in Eq. 3.6 guarantees that the penalty for violated constraints increases as r goes to zero while the upperbound on the exponent is needed to keep the minimum point $\vec{X}_r$ in the quadratic range of the penalty function. The quadratic and linear extensions to the penalty function are shown in Fig. 11 along with a conventional interior penalty function for comparison. A representation of the quadratic extended penalty function for different values of r is shown in Fig. 12.

3.5 Avoiding Negative Design Variables:

It is to re-emphasize that the extended penalty function resolves the difficulties arising from the fact that the ordinary interior penalty function is defined only in the feasible region. Even though extended interior penalty function is defined in the feasible region, it is

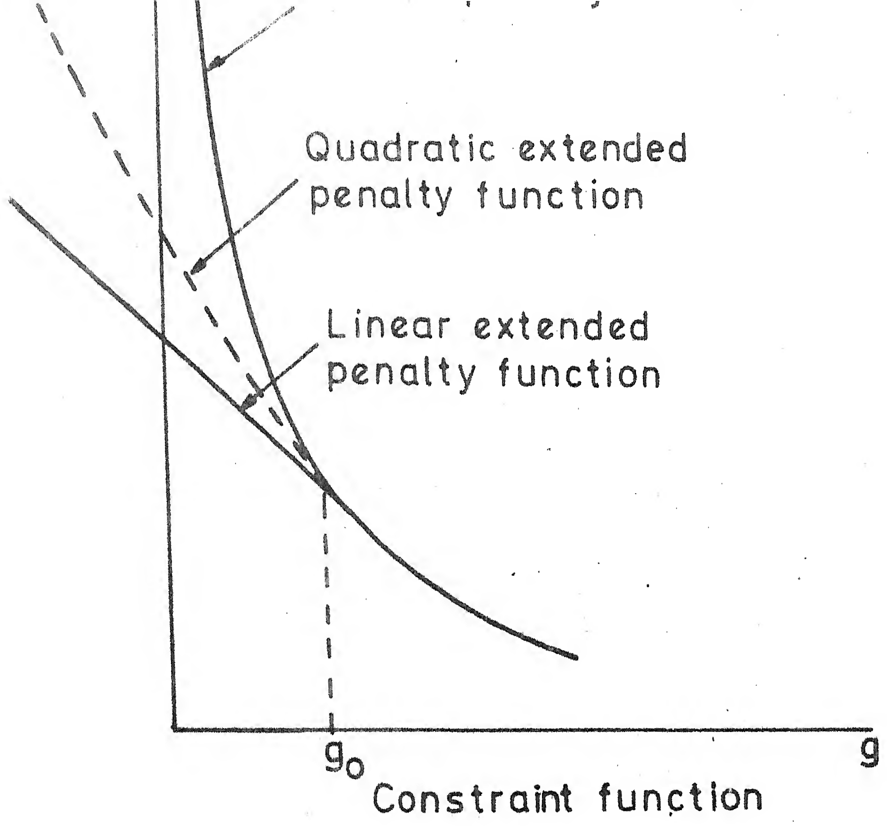
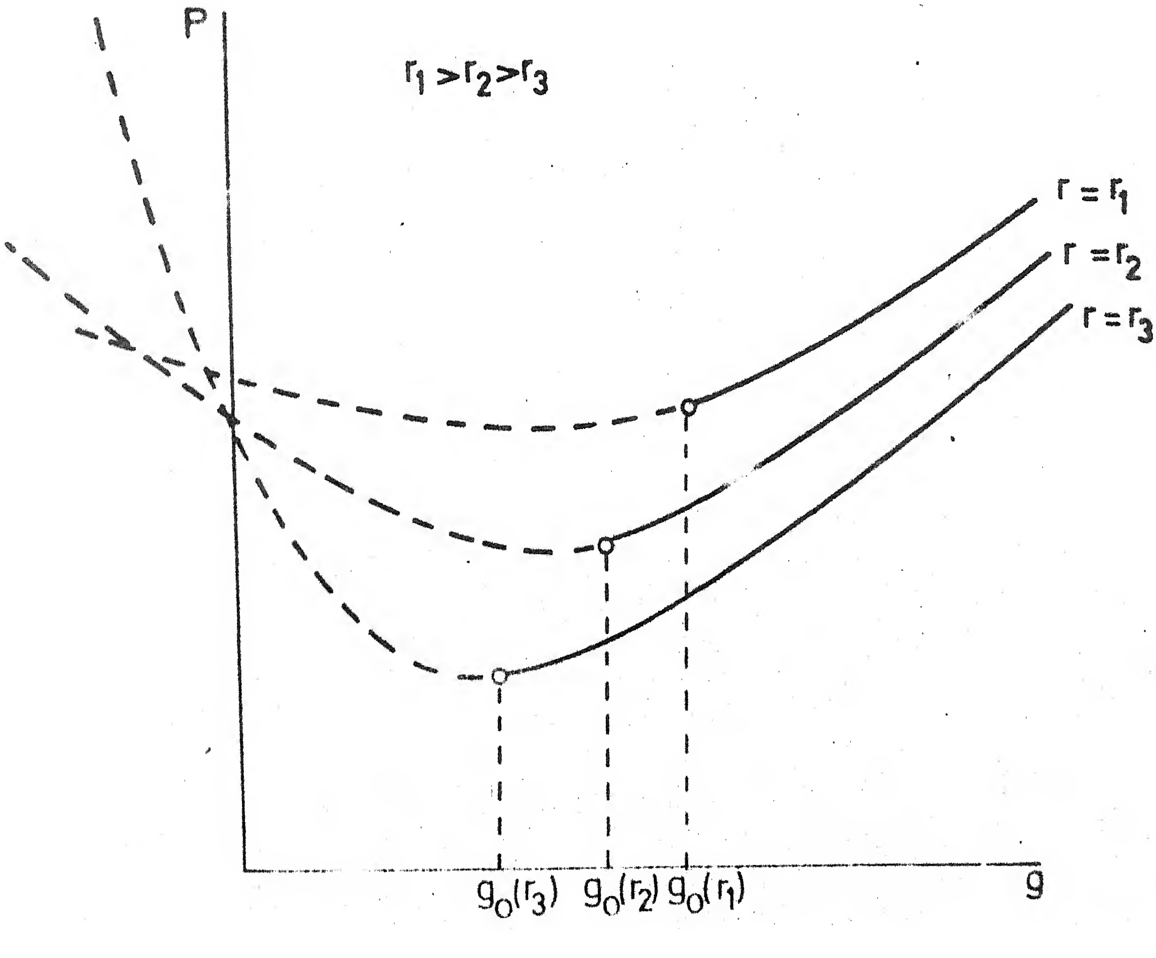


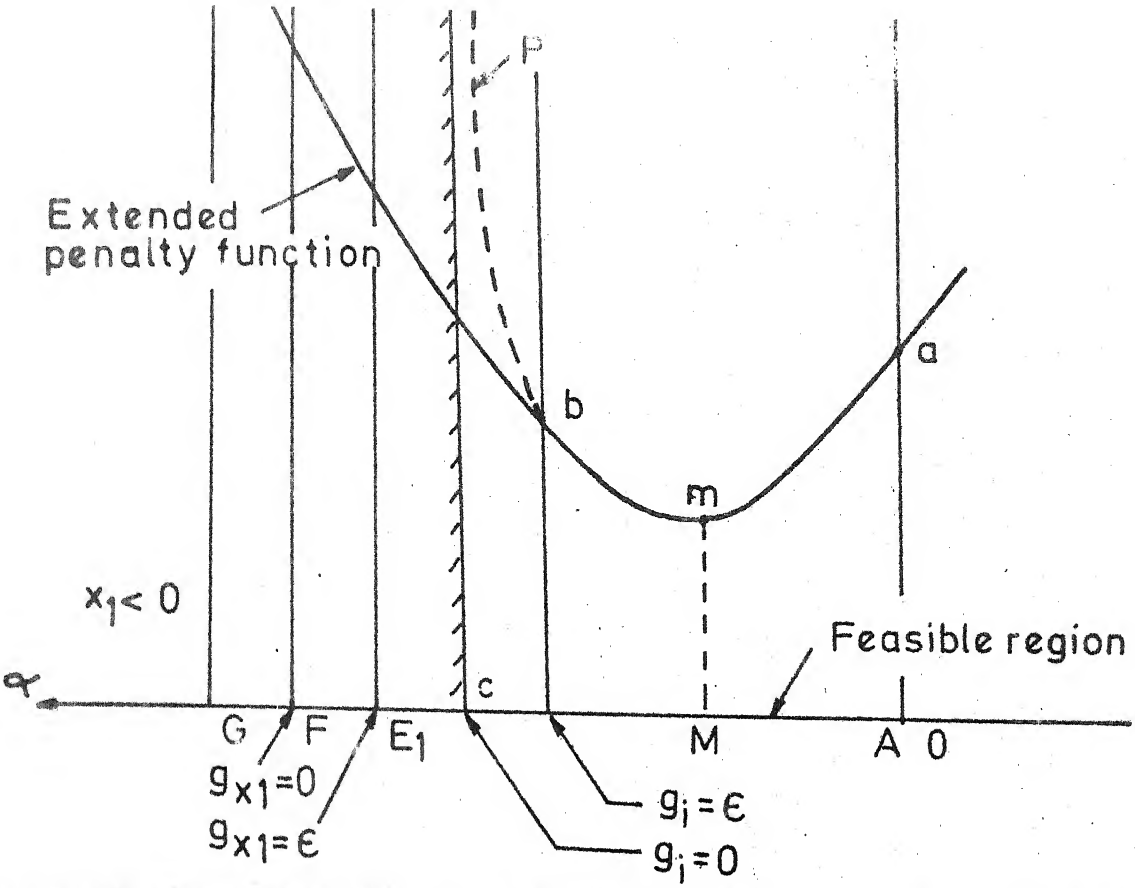
FIG.11 PENALTY FUNCTIONS



valid only over that portion of the design space where the constraints are meaningful. For example, g_j's loose their physical significance when one or more of the design variables takes on a negative value. Therefore if a search for the one-dimensional minimum leads to points with negative design variables then analysis should not be performed there. The search instead be returned to the portion of the design space where all the variables are positive. The method in |Ref. 13| , resolves this problem in one step, eliminating the time consuming task of making several trials. For completeness of presentation it is described herein.

Fig. 13 shows a one dimensional view of extended interior penalty function. The curve represents a plot of P for designs located along a straight line passing through the point A and having direction $\vec{S}$. For illustrative purposes the design space has been assumed two-dimensional, as shown in Fig. 14. There, the location of straight line in the direction $\vec{S}$ can be seen. Several points have been marked on this line in Figures 13 and 14, their meaning is described in the next paragraph.

Assume that starting from a point A in one dimensional minimization in the direction $\vec{S}$, determined by a certain



IG·13 ONE DIMENSIONAL MINIMIZATION ALONG A DIRECTION $\vec{S}$ STARTING FROM POINT A

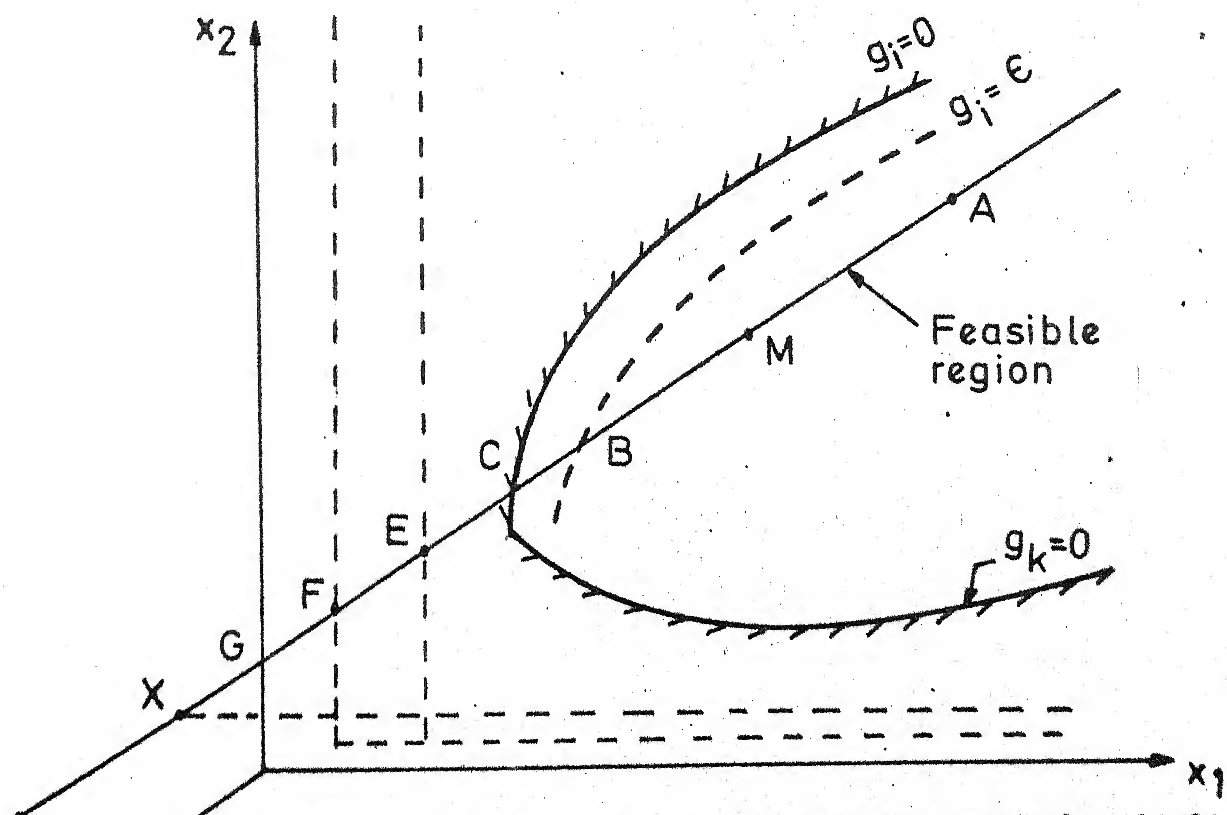


FIG.14 ONE DIMENSIONAL MINIMIZATION ALONG

algorithm, is to be performed. The starting point A is known and it is assumed to be the minimum point along the previous direction. The scalar variable α locates design points a long the direction $\vec{S}$. The scalar 'α' has its own origin at point 'A' and it is taken positive in the direction of decreasing P at point 'A' along $\vec{S}$. The minimum point sought along $\vec{S}$ is designated as point M. At point B along the direction $\vec{S}$ the constraint $g_i = \epsilon$ and therefore it defines the value of α corresponding to the transition point of extended interior penalty function. At point 'C' the constraint $g_i = 0$, therefore, this point is on the boundary separating the feasible region from the infeasible region. At point 'E_1' the lower bound side constraint corresponding to the design variable x_1 is equal to ϵ and E_2 has similar meaning with respect to x_2. At point F the side constraint related to x_1 equals to zero. Point 'G' where $x_1 = 0$, is the first point along the line for which P is not defined. At points X and Y the function P is undefined. Point X is characterized by the fact that $x_1 < 0$, $x_2 > 0$ while Y is located in the region where x_1 and x_2 are both negative.

Starting from A, the search seeks a point on the other side of M such that an interpolation function can be constructed and an approximate minimum may be determined

from it. If the step size is such that either X or Y is picked up, the analysis used to test whether or not the point is useful for the interpolation cannot be performed and another point has to be tried. For the present purpose, the points G, F, E_1, C and B, located on opposite side of M with respect to A are prospective candidates. Of these, point G can be excluded at the outset since it corresponds to $x_1 = 0$ and deletion of elements is not considered by current analysis. Furthermore, points B and C are difficult to locate because g_i is usually a complicated implicit function. Therefore, only points E_1 and F remain viable candidates.

It is straight forward task to locate either point E_1 or F because they represent the intersection of a straight line in direction $\vec{S}$ with hyperplane. However, E_1, has the advantage of being closen to M and therefore it is a better point at which to generate data for determining an approximate minimum. The detailed procedure for locating E_1 is now described. A lower bound side constraint function, related to a design variable x_j, is written as

$$g_{x1,j} = \frac{x_j - x_{alj}}{x_{alj}} = \cancel{\frac{x_j}{x_{alj}}} \; \frac{x_j}{x_{alj}} - 1 \quad (3.7a)$$

where x_{alj} represents the lower bound allowable value of the variable x_j.

At point E_j, $g_{x,j} = \epsilon$, so the jth co-ordinate of the point E_j can be evaluated from Equation (3.7a).

$$x_j^{(E_j)} = x_{a1j}(1+\epsilon) \tag{3.7b}$$

Now assume that starting from a feasible point A, a step size α_y is applied in the direction of $\vec{S}$ to get a point Y having some negative components. Let j^* denote the set of all subscripts corresponding to negative design variable components at point Y, and let 'm' be number of them. First, point Y is determined by

$$\vec{X}^{(Y)} = \vec{X}^{(A)} + \alpha_y \vec{S} \tag{3.7c}$$

Having identified all the design variables that are negative at Y, there will be 'm' points E_j ($j \in j^*$) to choose from, yet only one of them will satisfy the condition of belonging to the region where all the $x_i^{(E_j)}$,s are greater than zero. From Fig. (13) it is easy to see that the correct E_j is the one nearest to point A. As this distance is measured from A by the value α_E, the problem reduces to selecting

$$\alpha_{EK} = \min_{j \in j^*} \alpha_{Ej} \tag{3.7d}$$

After all α_{Ej} ($j \in j^*$) have been computed.

To obtain the α_{Ej}'s consider the relation.

$$\vec{X}^{(E_j)} = \vec{X}^{(A)} + \alpha_{Ej}\,\vec{S} \qquad (3.8a)$$

However, the jth component of $\vec{X}^{(E_j)}$ is known, as shown by Equation (3.7b) for this component equation (3.8a) implies

$$x_j^{(E_j)} = x_j^{(A)} + \alpha_{Ej}\,S_j \qquad (3.8b)$$

and equations (3.7b) and (3.8b) yield

$$\alpha_{Ej} = x_{alj}\,(1+\varepsilon) - x_j^{(A)} \qquad (3.8c)$$

After finding α_{Ek} from the condition expressed in equation (3.7d), the sought after point E_k is determined by

$$\vec{X}^{(E_k)} = \vec{X}^{(A)} + \alpha_{Ek}\,\vec{S} \qquad (3.8d)$$

Equations (3.8c), (3.7d) and (3.8d) show that it is simple to avoid design where the analysis variables are not defined.

3.6 Initial Value of r:

The initial value of the penalty parameter, r_o, has to be carefully chosen for better convergence. It is customary to choose r_o such that the objective function and the penalty term are of the same order or magnitude at the starting point. Thus,

$$r_o = \left| \frac{F\,(\vec{X})}{\Sigma\ 1/g_j} \right|$$

3.7 Gradient Evaluation:

Gradient for a quadratic extended penalty function is calculated as

$$\frac{\partial P}{\partial x_i} = \frac{\partial F}{\partial x_i} + r \sum_{j=1}^{ncon} \frac{\partial G_j(\vec{X})}{\partial x_i} \qquad (3.10)$$

$$\frac{\partial G_j}{\partial x_i} = - \frac{1}{g_j^2(\vec{X})} \times \frac{\partial g_j(\vec{X})}{\partial x_i} \qquad g_j(\vec{X}) \geq g_o$$

$$= \frac{1}{g_o^3} (2g_j(\vec{X}) - 3g_o) \times \frac{\partial g_j(\vec{X})}{\partial x_i} \qquad g_j(\vec{X}) \leq g_o$$

$$(3.11)$$

In the present work, partial derivatives , $\frac{\partial g_j(\vec{X})}{\partial x_i}$, are calculated by central difference method.

3.8 Hessian Matrix Development:

Hessian matrix (H) is the matrix of second derivatives of penalty function, P, and it is given by

$$H_{jk} = \frac{\partial^2 P}{\partial x_j \, \partial x_k} \qquad (3.12)$$

and

$$\frac{\partial^2 P}{\partial x_j \partial x_k} = \frac{\partial^2 F}{\partial x_j \partial x_k} + r \sum_{i=1}^{ncon} \frac{\partial^2 G_i}{\partial x_j \partial x_k} \qquad (3.13)$$

From Eqn. (3.5), for the quadratic extended penalty function, $\frac{\partial^2 G_i}{\partial x_j \partial x_k}$ is given by

$$\frac{\partial^2 G_i}{\partial x_j \partial x_k} = g_i^{-3} \left[2 \frac{\partial g_i}{\partial x_j} \frac{\partial g_i}{\partial x_k} - g_i \frac{\partial^2 g_i}{\partial x_j \partial x_k} \right]$$

$$\text{for } g_i \geq g_o \qquad (3.14)$$

$$= g_o^{-3} \left[\frac{2 \partial g_i}{\partial x_j} \frac{\partial g_i}{\partial x_k} + g_o \left(\frac{2 g_i}{g_o} - 3 \right) \frac{\partial^2 g_i}{\partial x_j \partial x_k} \right]$$

$$\text{for } g_i \leq g_o \qquad (3.15)$$

Because of the factors g_i^{-3} and g_o^{-3} in Eqs. (3.14, 3.15), the main contribution to the second derivatives of the penalty function is from the constraints which are nearly critical. For these constraints the following approximations for the second derivatives of the quadratic extended penalty function may be used,

$$\frac{\partial^2 G_i}{\partial x_j \, \partial x_k} = 2 \, g_i^{-3} \frac{\partial g_i}{\partial x_j} \frac{\partial g_i}{\partial x_k} \quad \text{for } g_i \geq g_o$$

$$(3.17)$$

$$= 2 g_o^{-3} \frac{\partial g_i}{\partial x_j} \frac{\partial g_i}{\partial x_k} \quad \text{for } g_i \leq g_o$$

It is noteworthy that this approximation requires only the first derivatives of the constraints. Therefore, the computational effort needed to generate the Hessian Matrix shall

be much less.

3.9. Modified Newton's Method:

3.9.1 Introduction:

The unconstrained minimization of the penalty function, $P(\vec{X},r)$, is performed by modified Newton's method in the present work. The disadvantage of the quasi-Newton algorithms is that the number of function evaluations required for the optimization procedure (which is measure of computational efficiency of the algorithm) is a linear function of the number of design variables. These algorithms, therefore, are not suitable for use when a large number of design variables are involved. It was reported in Ref. |18| , that generalized Newton's method applied with an interior penalty function formulation can be used to overcome this disadvantage because it is possible to obtain a simple approximation to the second derivatives of the penalty function necessary for generalized Newtons method. This method requires a smaller number of function evaluations and is independent of the number of design variables. It is, therefore, useful for solving optimum structural design problems having a large number of design variables. However, in the present work, Newton's method with modification is used for unconstrained minimization

with an intention to develop a general purpose computer programme to handle large problems as is common in structural analysis even though, due to decomposition principles applied, the problem in the present work does not have large number of design variables.

There are two theoretical arguments against the generalized Newton's method:

(i) If Hessian matrix (H) is not a positive definite matrix, a move in the direction given by, $-[H]^{-1}\nabla P$, may result in an increase rather than a decrease in $P(\vec{X},r)$, yielding, $\alpha_i=0$, and terminate the process at the starting point $\vec{X}_i$.

(ii) The Hessian matrix may not have an inverse, even if $P(\vec{X},r)$ is convex.

The modified second-order method used in the present work takes into account these limitations of the generalized Newton method. The direction vector $\vec{S}$ is generated according to the algorithm given below. The basis for the algorithm is given in Ref. |8| reproduced herein for the purposes of completeness of presentation.

The direction vector $\vec{S}$ is generated according to two rules. In both cases $\vec{X}_{i+1} = \vec{X}_i + \alpha_i \vec{S}$, where α_i is chosen to be the smallest value of $\alpha \geq 0$ for which

$\vec{X}_i + \alpha_i \vec{S}$ is a local minimum of $P(\vec{X}_{i+1}, r)$.

The rules are as follows:

(i) If H has negative eigenvalue, let $\vec{S}$ be a vector where

$$\vec{S}^T [H] \vec{S} < 0 \quad \text{and} \quad \vec{S} \nabla P \leq 0 \qquad (3.18)$$

(ii) If H has all eigenvalues greater than or equal to zero, choose $\vec{S}$ such that

$$\text{either} \quad [H] \vec{S} = 0 , \quad \vec{S}^T \nabla P < 0 \qquad (3.19)$$

$$\text{or} \quad [H] \vec{S} = -\nabla P \qquad (3.20)$$

in which case, if $\nabla P \neq 0$, it can be proved that $\vec{S}^T [H] \vec{S} = - \vec{S}^T \nabla P > 0$. Both Eqs. (3.19) and Eq. (3.20) can not hold. Let $\vec{S}$ satisfy (3.19) or (3.20).

The only case in which Rules 1 and 2 fail to generate a non zero direction $\vec{S}$ is when $[H]$ is a positive semidefinite matrix and $\nabla P = 0$; that is, a point that satisfies the first-and second-order necessary conditions that $\vec{X}_i$ be a local constrained minimum of $P(\vec{X}, r)$.

The rationale for Rule 1 is that if the second derivative matrix has a negative eigenvalue there are certain directions along which the function $P(\vec{X}, r)$ decreases and along which the rate of decrease also decreases. That is, for a vector $\vec{S}$ satisfying Eq. (3.18),

$$d\,P\,(\vec{X}_{i+1},r)\,/d\alpha \;=\; \vec{S}^{T}\nabla P \;\leq 0 \quad \text{and}$$

$$d^2\,P\,(\vec{X}_{i+1},r)\,/\,dx^2 \;=\; \vec{S}^{T}\,[H]\,\vec{S} \;<0 \tag{3.21}$$

$$\text{at} \quad \alpha = 0$$

Thus, unless P varies along the ray $\vec{S}$ emanating from $\vec{X}_i$ so that eventually $d^2\,P\,(\vec{X}_{i+1},r)/\,d\alpha^2 \;\geq 0$, the function value will tend to $-\infty$. This rule provides a guide for locating a region where [H] is positive semidefinite. For the minimization of a nonconvex function a few iterations along vectors satisfying Eq. (3.18) can result in significant progress in minimization algorithm.

The same remarks apply to (3.19) of Rule 2 . If $[H]_{\text{at } \vec{X}_{i+1}}$, $\vec{S} = 0$ holds for α indefintely large, $P(\vec{X}_{i+1},r)$ will tend to $-\infty$, Finally, if [H] is positive definite , Eq. (3.20) yields the Newton vector, when Eq. (3.20) is used, in general,

$$\vec{S} \;=\; -\,[H]^{-1}\nabla P \tag{3.22}$$

It is to be noted that any $\vec{S}'$ satisfying Eq. (3.21) has the desirable property that if P $(\vec{X},r)$ is a positive semidefinite quadratic form having a finite unconstrained minimum.

Rules 1 and 2, although quite useful computationally, are not guaranteed to generate a sequence of points having

limit points satisfying the first- and second order necessary conditions for a minimum. A class of direction vectors for which any existing limit point has this property is established in Ref. |21| one particular example of this type of direction vector is

$$\vec{S} = -A \nabla P + \delta^i e^i \tag{3.23}$$

where A is a positive definite matrix, e^i is an eigen vector of [H] with minimum eigenvalue, and δ^i is a scalar,

where

$$\delta^i = \begin{cases} -1 & \text{if } [H] \text{ has a negative eigenvalue and } P^T e^i \geq 0 \\ 0 & \text{if } [H] \text{ is positive semi-definite} \\ 1 & \text{if } [H] \text{ has a negative eigenvalue and } P^T e^i < 0 \end{cases}$$

From the above observations, the suggested algorithm is given as follows.

3.9.2 Search Direction for Modified Newton's Method:

The search direction $\vec{S}$ for one-dimensional minimization after developing Hessian matrix [H] is given by the following algorithm.

(i) Reduce the Hessian matrix [H] to the form

$$H = L D L^T \tag{3.24}$$

where L is a nonsingular lower triangular matrix with units on the main diagonal and D is a diagonal matrix.

(ii) If D has all positive diagonal elements, solve for

$$\vec{S} = - [H]^{-1} \nabla P \qquad (3.25)$$

(iii) If D has some diagonal elements that are negative, solve $L^{T} \vec{T} = \vec{A}$ where $\vec{A}$ is a column vector with jth component = 0 if the jth diagonal element of D > 0 and with jth component = 1 if the jth diagonal element of D $\leq$ 0 . Let $\vec{S} = \vec{T}$ if $\vec{T}^T \nabla P \leq 0$, and $\vec{S} = - \vec{T}$ otherwise.

(iv) If D has all nonnegative diagonal elements, and at least one is zero, the vector $\vec{S}$ is generated according to $\vec{S} = - [H]^{-1} \nabla P$.

3.9.3 <u>Algorithm for Step Length for Modified Newton's Method</u>:

Quadratic interpolation method has been used to find the minimum of P $(\vec{X}, r)$ in a direction $\vec{S}$. This one-dimensional minimization method finds the minimizing step length α^* in two stages. In the first stage, the $\vec{S}$- vectpr is normalized so that a step length of $\alpha = 1$ is acceptable. In the second stage, the function P $(\vec{X}, r)$ is approximated by a quadratic function $h(\alpha)$ and the minimum $\bar{\alpha}^*$, of $h(\alpha)$ is found. If $\bar{\alpha}^*$ is not sufficiently closer to the minimum α^*, the third stage is used. In this stage, a new quadratic function $h'(\alpha) = a' + b'\alpha + c'\alpha^2$ is used to approximate

$P(\vec{\lambda},r)$, and a new value of $\bar{\alpha}^*$ is found. This procedure is continued until a α^* , which is sufficiently close to α^* is found. The algorithm of this method is well-documented in text books and is discussed in detail in Ref. |12, 15|.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction:

An illustrative example using the techniques of optimum design described in previous chapters is discussed herein. The problem, that was solved in Ref. |2| , and later in Ref. |11| has been taken up here too for the sake of comparison. The results are presented and discussed and conclusions are drawn.

All computational work has been carried out in DEC 1090 system at IIT, Kanpur.

4.2 The Example Problem:

The design of a 80 m high ventilated R.C. Chimney is considered here. The relevant data are given below:

(1) Geometry:

Height of chimney = 80 m

Type - lined and ventilated

Ventilated air space = 10 cm

Thickness of fire brick lining = 11.5 cm

The lining is supported at corbels at 8m intervals

Mean diameter of shell at top = 4.85 m

Shell thickness at top = 15 cm

Minimum vertical reinforcement percentage throught the chimney shell = 0.3

Flue opening of 3m x 3m at 8 m from base.

(2) Temperature:

Temperature of flue gas = $160^\circ C$

Minimum temperature of atmosphere = $4^\circ C$

r_q = 0.5

C_c = 1.488 $K_{cal}/m/h/^\circ C$

C_b = 1.25 $K_{cal}/m/h/^\circ C$

K_1 = 50.00 $K_{cal}/m^2/h/^\circ C$

K_2 = 58.59 $K_{cal}/m^2/h/^\circ C$

K_s = 10.20 $K_{cal}/m^2/h/^\circ C$

Coefficient of linear expansion for concrete

$L = 11 \times 10^{-6}$ per $^\circ C$

(3) Wind Pressure:

Basic wind pressure

100 kg/m^2 from 0 m to 24 m

115 kg/m^2 from 24 m to 56 m

122 kg/m^2 from 56 m to 80 m

Shape factor = 0.7

(4) Earthquake |3|

Damping factor = 0.02

Factor F_o = 1.0

Importance Factor I = 1.5

(5) Material Properties:

Cube strength of concrete(f_c)	= 250 kg/cm^2
Yield stress of mild steel (f_{sy})	=2600 kg/cm^2
Unit weight of concrete	=2400 kg/m^3
Unit weight of fire-brick lining	=2100 kg/m^3
Young's modulus of mild steel	=2.0x10^6 kg/cm^2
Young's modulus of concrete	=181818.18 kg/cm^2
Modular ratio	= 11

(6) Material Costs:

Cost of concrete shell (C_u)	= Rs. 180 per m^3
Ratio of unit cost of steel and concrete (S)	= 80

(7) Permissible Stresses:

For M 250 concrete and mild steel as reinforcement the permissible stresses for various load conditions have been computed and are given in table 2 below.

TABLE 2

PERMISSIBLE STRESSES

Load conditions	Permissible stresses(kg/cm^2)	
	Concrete	Steel
D L. + W.L.	95.0	1480
D.L. + E.Q.	100.0	1560
D.L. + Temp.	82.5	1430
D.L. + W.L. + Temp.	125.0	1690
D.L. + E Q. + Temp.	125.0	1690

4.3 Note on Computer Program:

The chimney has been divided into 10 stages. In general the stage boundaries are taken at corbel points. A computer program in FORTRAN is developed to solve the present problem, and the same is reproduced in Appendix A. The computer program consists of one main program and ten subroutines. The main program initiates the program execution and prints the values of optimum design variables at each decomposed stage of the chimney. The subroutines are briefly explained below:

SUMT : This subroutine performs the constrained minimization using extended interior penalty function method.

NEWTON : This segment is called by SUMT and performs the unconstrained minimization of the penalty function using Modified Newton's method.

QUAD : This module is called by Newton for performing one dimensional minimization. It uses Quadratic interpolation method.

GRAD : In this the gradient of the penalty function is evaluated. It is called by NEWTON.

HESSI : In this subroutine the search direction for linear minimization is generated after evaluating

Hessian matrix. It is called by NEWTON.

ANALYS : This segment does the analysis of the chimney and calculates the values of the constraints.

ANEG : This subroutine eliminates the negative design variables before any analysis is performed.

MATINV : This module finds the inverse of the Hessian matrix and is called by HESSI.

PENALT : This subroutine evaluates the penalty term of the penalty function.

ANEUT : It gives the value of β (Neutral axis of the chimney) by solving the transcendental equation (2.21).

4.4 Pertinent Values of Parameters used in the Computer Programme:

The initial value of transition parameter (g_0) is taken as 0.1 and the exponent (p) of the penalty parameter (r) is taken as 0.5. The initial value of r is chosen such that the penalty term equals the value of the objective function. After each unconstrained minimization r is reduced by a factor of 10.

One dimensional searches are conducted by using a parabolic approximation to the penalty function along a search direction. This approximation is based on the three best design points previously determined when any of the three criteria are satisfied: (1) 20 steps have been taken in the search direction : (2) the expected improvement in the object function $P(\vec{X},r)$ based on the parabolic approximation is less than 0.05 percent; or (3) the step size is reduced to less than 0.01 times the starting step size. The starting step size is taken as 0.1 in each one dimensional search.

Convergence Criteria:

In the present problem the iterative search process is terminated according to the following convergence criteria.

(i) For each unidirectional search

$$\left| \frac{P''(\vec{X}^*, r) - P'(\vec{X}^* r)}{P'(\vec{X}^*, r)} \right| \leq 0.0001$$

$$|\vec{S} \nabla P| \leq 0.001$$

(ii) For unconstrained minimization

$$\left| \frac{P''(\vec{X}, r) - P'(\vec{X}, r)}{P'(\vec{X}, r)} \right| \leq 0.005$$

(iii) For final completion of the search

$$|x_i'' - x_i'| \leq 0.01$$

$$\left| \frac{F''(\vec{X}) - F'(\vec{X})}{F'(\vec{X})} \right| \leq 0.0001$$

In the above criteria the notations '' and ' refer to two consecutive points of comparison.

4.5 Results:

The results are obtained for each decomposed optimization problem and they are compared with those in Ref. |2, 11| and are tabulated in Table 3. Only mean diameter and wall thickness of the chimney are tabulated in Table 3 for each case, since the percentage of reinforcement at each cross section attains the minimum permissible value of 0.3 percent at each optimum in all the three cases.

TABLE 3

COMPARISON OF OPTIMUM DESIGN WITH THOSE REPORTED IN REF. |2,11|

Elevation in m	A			B			C		
	Optimum design by present formulation			Optimum design of Ref.2			Optimum design of Ref. 11		
	Mean dia. in m	Thickness in m	Cost in Rs.	Mean dia. in m.	Thickness in m	Cost in Rs.	Mean dia. in m.	Thickness in m	Cost in Rs.
80	4.85	0.15	-	4.85	0.150	-	4.85	0.15	-
72	4.85	0.15	4012	4.86	0.151	4048.4	4.85	0.15	4012
64	4.85	0.15	4012	4.86	0.151	4107.7	4.85	0.15	4012
56	4.85	0.15	4012	4.86	0.151	4093.3	4.85	0.15	4012
48	4.85	0.15	4012	4.86	0.151	4032.1	4.85	0.15	4012
40	4.85	0.15	4012	4.86	0.151	4046.5	4.85	0.15	4012
32	4.85	0.15	4012	4.86	0.151	4039.8	4.85	0.15	4012
24	4.85	0.15	4012	4.86	0.151	4057.2	4.85	0.15	4012
16	4.85	0.15	4012	4.86	0.152	4060.6	4.85	0.15	4012
8	6.04	0.15	4673	5.86	0.172	4698.4	6.08	0.15	4702
0	6.04	0.15	4972	5.86	0.154	5011.8	6.08	0.15	5003
Total cost of superstructure in (Rs.)	41741			42196			41801		
Computer time	40 sec. (DEC 1090)			10 min (approx.)			5 min. (WATFOR)		

4.6 Discussions:

It can be seen that at the optimum design, the thickness of the shell at all sections reaches the lower bound value and so does the value of the percentage of reinforcement at each cross section. The mean diameter of the chimney from the top upto the flue opening remains constant, and it increases to a larger diameter at the flue opening and remains same upto the base. This has been consistently obtained for this structure by using other optimization procedures in Ref. |2,11|.

At all the cross sections, the active constraints are the side constraints on thickness and mean diameter. The most critical section, for the chimney considered, is obviously the section at the flue opening. The behaviour constraints, viz., compressive stress in concrete due to dead and lateral loads, and due to dead load, lateral load and temperature, were observed to be critical at the flue opening.

The computation time required to solve the present problem is 40 seconds in DEC 1090 system. It took 5 minutes of CPU time Ref. |11| and 10 minutes of CPU time Ref. |2| for the same problem on IBM 7044. But it is believed that the present computer (DEC 1090) is faster by a factor of 4 to 5 as compared to IBM 7044. Had this present problem been done on IBM 7044, it would have taken 200 secs (40 x 5), that is approximately $3\frac{1}{2}$ minutes. So it can be concluded that the present technique to find the optimum solution is definitely superior to those used in Ref. |2,11|, for solving this class of problems.

The total minimum cost obtained in the present work as well as in Ref. |2,11| are almost the same.

Lastly but not the least, the penalty-function technique seems to be gaining increasing popularity in the field of structural optimization, and the volume of literature on the practical applications in this field is increasing rapidly. The modified Newton's method for unconstrained minimization in the penalty function formulation, is robust in the sense it works well though the Hessian matrix is not positive definite. And also with the inclusion of extended penalty function technique, the method is able to accept infeasible starting points. This is a quite important feature when dealing with complex design problems for which feasible starting points are not readily available. Although the method is already used with success in practical design work, there are great potentials for further improvements.

4.7 Suggestions for Further Work:

(i) Cost of foundation could be included in the objective function along with the present objective for the minimum cost design of R.C. chimney.

(ii) Aerodynamic effects on the chimney should be considered for more realistic analysis of the structure.

(iii) Cost of brick lining as well as cost of shuttering could also be included in the objective function.

REFERENCES

1. I.S. 4998 (Part 1) 1968, Criteria for Design of Reinforced Concrete Chimneys, ISI, New Delhi, 1975.

2. Kapoor, M.P. and Hariharan, M., 'Optimal Design of Reinforced Concrete Chimneys', IUTAM Symposium on Optimization in Structural Design, Warsaw/Poland, 1973, pp. 395-404.

3. IS 1893-1975: Criteria for Earthquake Resistant Design of Structures, ISI, New Delhi 1975.

4. IS 456-1964: Code of Practice for Plain and Reinforced Concrete, ISI, New Delhi.

5. IS 2950 (Part 1)- 1973: Code of Practice for Design and Construction of Raft Foundation, ISI, New Delhi, 1973.

6. Timoshenko, S., and Woinowsky- Krieger, S., 'Theory of Plates and Shells', McGraw-Hill, New York, 1959.

7. Bowles, J.E., 'Foundation Analysis and Design', McGraw-Hill, 1968.

8. Fiacco, A.V., and McCormick, G.P., 'Nonlinear Programming-Sequential Unconstrained Minimization Technique', John Wiley, 1968.

9. Taylor, C.E. and Turner, L., 'Reinforced Concrete Chimneys', Butler and Tanner Ltd., France and London, 1960.

10. IS: 875-1964, 'Code of Practice for Structural Safety of Buildings: Loading Standards', ISI New Delhi,1975.

11. William S.LA Londe, Jr., 'Concrete Engineering Handbook', MC Graw-Hill, 1961.

12. Bandyopadhyay, N., 'Integrated Optimum Design of R.C. Chimney', M.Tech. Thesis, July, 1976, IIT Kanpur.

13. Fox, R.L., 'Optimization Methods for Engineering Design', Addison-Wesley, Reading, Massachussetts, 1971.

14. Casis, J.H. and Schmit, L.A., 'On Implementation of the Extended Interior Penalty Function', Int. J.Num. Meth. Engng., 10(1), p.3-23, (1976).

15. Haftka, R. T. and Starnes, J.H., 'Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization', AIAA Journal, Vol.14, No.6, pp. 718-724, June 1976.

16. Rao, S.S., 'Optimization Theory and Applications', Wiley Eastern, 1979.

17. Gallagher, R.H. and Zienkiewicz, O.C., 'Optimum Structural Design', John Wiley, 1973.

18. Bathe, Klans-Jurgen and Wilson, E.L., 'Numerical Methods in Finite Element Analysis', PHI, 1978.

19. Haftka, R.T., 'Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements', TN-D-7264, July 1973, NASA.

20. Carroll, C W , 'The Created Response Surface Technique for Optimizing Non-linear Restrained Systems', Operations Research, Vol.9, pp. 169-184, 1961.

21. Kavlie, D. and Moe, J., 'Automated Design of Frame Structures', ASCE Journal of Structural Div., Vol. ST1, Jan. 1971, pp. 33-62.

22. Mc Cormick, G.P., and W.I. Zangwill, 'A Technique for Calculating Second Order Optima', Technical paper, Research Analysis Corp., McLean, Virginia.

APPENDIX-A

```
C     ************************************************************
C     PROGRAM TO MINIMIZE CONSTRAINED FUNCTION
C     USING SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE
C     NTYPN-TYPE OF UNCONSTRAINED MINIMIZATION TECHNIQUE
C     NTYPL- TYPE OF ONE DIMENSIONAL MINIMISATION TECHNIQUE
C     C1=INITIAL VALUE OF R
C     C2=REDUCTION FACTOR FOR FOR R
C     NTYPE-TYPE OF EXTENDED INTERIOR PENALTY FUNCTION TECHNIQUE
C     NTYPE=1   LINEAR EXTENDED INTERIOR PENALTY FUNCTION
C     NTYPE=2  QUADRATIC EXTENDED INTERIOR PENALTY FUNCTION
C     NTYPE=3  CUBIC EXTENDED INTERIOR PENALTY FUNCTION
C     STLMT-MINIMUM STEP LIMIT
C     ****************************************************************
      DIMENSION DES(3),DAL(3),CONS(10)
      COMMON/METHOD/NTYPN,NTYPL,NTYPE
      COMMON/AREA4/W,AMOM,ALP,WO,WC,WCO
      COMMON/AREA2/TOTL,TLIN,ATRGAP
      COMMON/ET/AH
      COMMON/AREA7/ODBO(11),OTH(11),SABYG(30)
      COMMON/CONS1/NCON,NVAR
      COMMON/QUAD1/STEP,EPS1,EPS2,MAXFIT,STLMT
      COMMON/STAGE/NSTAGE,NTIM,NSO
      COMMON/AREA5/P,PM,BETA,EBYR
      COMMON/ALLOW1/DAL
      COMMON/CONVE/ER1,ER2
C     COMMON/CUB1/STEP,EPS1,EPS2,MAXFIT,STLMT
      COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
      COMMON/AREA9/CONS
      COMMON/PARAM/PI,COSRA,AM,COC
      PI=3.1415926
      READ(20,*)NTYPN,NTYPL,NTYPE,NSO
      READ(20,*) NVAR,NCON,ER1,ER2
      READ(20,*) C1,C2
      READ(20,*) STEP,MAXFIT,EPS1,EPS2,STLMT
      READ(20,*) (DES(I),I=1,N)
      READ(20,*) (DAL(I),I=1,N)
      READ(20,*) (SABYG(I),I=1,30)
      READ(20,*)COSRA,AM,COC
      DO 100 I=1,10
      TYPE 10,I
10    FORMAT(/10X,'NSTAGE:',I2/)
      NSTAGE=I
      NTIM=0
      IF(NSTAGE.LE.10) AH=8.
      CALL ANALY(DES)
      TYPE 15,TOTL
15    FORMAT(/,10X,'THE PRESENT SECTION IS AT A DISTANCE OF',F6.1,
     1' METRES FROM BASE'/)
      TYPE 17,W,AMOM
17    FORMAT(/,10X,'WEIGHT OF THE CHIMNEY ABOVE THE SECTION',F10.2,/,
     110X,'MOMENT AT THE PRESENT SECTION',F14.2,/)
      TYPE 18,(CONS(K),K=1,NCON)
18    FORMAT(/,5X,'VALUES OF CONSTRAINTS AT STARTING POINT',/,5X,
     110(1X,F9.6)/)
      TYPE 250,EBYR,BETA,P,ALP
250   FORMAT(/10X,'EBYR=',F5.2,10X,'BETA=',F5.2,10X,'P=',F6.4,10X,
     1'ALPA=',F10.4/)
      TYPE 16,WO,ALP
```

```
10     FORMAT(/10X,'WEIGHT OF THE CHIMNEY ABOVE THE PREVIOUS SECTION:',
      1F14.4,/,10X,'ALP:',F8.3/)
       CALL SUMT(DES)
       TYPE 20,(DES(J),J=1,3)
20     FORMAT(/10X,'D1=',F10.4,10X,'D2=',F10.4,10X,'D3=',F10.4/)
       DDBD(I)=DES(1)
       DTH(I)=DES(2)
       WO=W
       WCO=WC
100    CONTINUE
200    STOP
       END
```

```
        SUBROUTINE SUMT(DES)
        DIMENSION DES(3),Y(3),CONS(10)
        COMMON/METHOD/NTYPH,NTYPL,NTYPE
        COMMON/STAGE/NSTAGE,NTIM,NSQ
        COMMON/AREA9/CONS
        COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
        COMMON/CONS1/NCON,N
        COMMON/CONVE/ER1,ER2
        COMMON NSEQ
        COMMON/PHI/PHINEW
00      R1=C1
        K=1
        NSEQ=K
        CALL PENALT(DES,FCONS)
        TYPE 1,FCONS
        FORMAT(/10X,'FCONS=',E14.8/)
        FX=OBJFN(DES,N)
        R1=C1*FX/(SIGMA)
        FCONS=SIGMA*R1
10      FM=FX+FCONS
        TYPE 300, (DES(I),I=1,N),FX
        TYPE 550, FM
50      FORMAT(5X,'MODIFIED OBJECTIVE FUNCTION VALUE=',F15.8)
00      FORMAT(5X,'STARTING VALUES OF THE VARIABALES:',3(5X,F12.5)/,
       15X,'OBJECTIVE FUNCTION VALUE:',F12.5//)
00      FORMAT(5X,'OPTIMUM VALUES OF THE VARIABLES:',3(5X,F12.5)/,
       15X,'OBJ FN VALUE:',F12.5/)
        TRAP=0.1
        COD=TRAP/(R1**0.5)
75      DO 600 I=1,N
00      Y(I)=DES(I)
        FY=FX
        CALL NEWTON(DES)
        FX=OBJFN(DES,N)
        CALL PENALT(DES,FCONS)
        FN=FX+FCONS
        TYPE 675,K,FN,FX,(DES(I),I=1,N)
75      FORMAT(2X,'SEQUENCE NO:',I2,2X,'PHI FN VALUE:',E14.8,2X,'OBJ
       1 FN VALUE:',F12.6,2X,'DES(1)=',F8.4,2X,'DES(2)=',F8.4,2X,
       2'DES(3)=',F8.4//)
        TYPE 650, (CONS(I),I=1,NCON)
50      FORMAT(5X,'NORMALISED CONSTRAINT FN VALUES=',/,10(1X,F9.6)/)
        TYPE 660, R1,K,FCONS
60      FORMAT(5X,'R VALUE:',F14.9,'...........(',I2,')',
       1     /,5X,'PENALTY FUNCTION VALUE   =',F12.4)
        ERRMAX=(DES(1)-Y(1))/Y(1)
        ERRMAX=ABS(ERRMAX)
        DO 700 I=2,N
        TEMP=(DES(I)-Y(I))/Y(I)
        TEMP=ABS(TEMP)
        IF(TEMP.GT.ERRMAX) ERRMAX=TEMP
00      CONTINUE
        RELERR=ABS(FY-FX)/FY
        PHIERR=ABS((FM-FN)/FN)
```

```
      IF((RELERR.LE.ER1.AND.ERRMAX.LE.ER2).OR.(K.GE.NSO)) GO TO 310
 05   FM=FN
      K=K+1
      R1=C2*R1
      TRAP=COD*R1**0.5
      GO TO 575
      GO TO 575
 10   IF((PHIERR.LE.0.001).OR.(K.GE.NSQ)) GO TO 320
      GO TO 305
 20   TYPE 400, (DES(I),I=1,N),FX
 00   RETURN
      END
```

```
      SUBROUTINE PENALT(DES,FCONS)
C     THIS SUBROUTINE CALCULATES THE PENALTY TERM
C     OF THE PENALTY FUNCTION
      DIMENSION DES(3),G(3),CONS(10)
      COMMON/CONS1/NCON,NVAR
      COMMON/AREA9/CONS
      COMMON/METHOD/NTYPN,NTYPL,NTYPE
      COMMON/STAGE/NSTAGE,NTIM,NSQ
      COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
      COMMON NSEQ
      IF(NTIM.GE.1) GO TO 20
      IF (NSEQ.EQ.1) GO TO 7
      DO 5 I=1,NCON
      IF(CONS(I).GE.TRAP) GO TO 20
      CONTINUE
      SIGMA=0.0
      DO 10 I=1,NCON
      IF(CONS(I).LT.0.0001) GO TO 10
      SIGMA=SIGMA+1.0/CONS(I)
      CONTINUE
      FCONS=SIGMA*R1
      RETURN
      GO TO(25,40,60),NTYPE
C     LINEAR EXTENDED INERIOR PENALTY FUNCTION
      SIGMA=0.0
      DO 30 I=1,NCON
      IF(CONS(I).GE.TRAP) GO TO 35
      SIGMA=SIGMA+(2.0-CONS(I)/TRAP)/TRAP
      GO TO 30
      SIGMA=SIGMA+1.0/(CONS(I))
      CONTINUE
      FCONS=SIGMA*R1
      RETURN
C     QUADRATIC EXTENDED INTERIOR PENALTY FUNCTION
      SIGMA=0.0
      DO 50 I=1,NCON
      IF(CONS(I).GE.TRAP) GO TO 45
      UT=CONS(I)/TRAP
      SIGMA=SIGMA+((UT-1.0)**2-UT+2.0)/TRAP
      GO TO 50
      SIGMA=SIGMA+1.0/(CONS(I))
      CONTINUE
      FCONS=SIGMA*R1
      RETURN
C     CUBIC EXTENDED INTERIOR PENALTY FUNCTION
      SIGMA=0.0
      DO 70 I=1,NCON
      IF(CONS(I).GE.TRAP) GO TO 65
      UT=CONS(I)/TRAP-1.0
      SIGMA=SIGMA+((UT**3)+UT*UT-UT+1.0)/TRAP
      GO TO 70
      SIGMA=SIGMA+1.0/(CONS(I))
      CONTINUE
      FCONS=SIGMA*R1
      RETURN
      END
```

```
SUBROUTINE NEWTON(DES)
****************************************************************
UNCONSTRAINED MINIMIZATION TECHNIQUE
USING NEWTON'S METHOD WITH QUADRATIC  FOR ONE DIMENSIONAL
MINIMISATION
****************************************************************
DIMENSION DES(3),G(3),S(3),HES(3,3),DG(10,3)
COMMON/CONS1/NCON,N
COMMON/QUAD1/STEP,EPS1,EPS2,MAXFIT,STLMT
COMMON/CUB1/STEP,EPS1,EPS2,MAXFIT,STLMT
COMMON/STAGE/NSTAGE,NTIM,NSO
COMMON NFEVA
COMMON PHINEW
NFEVA=0
NTIM=0
NTIM=NTIM+1
TYPE 85,NTIM
FORMAT(/,5X,'NO. OF CYCLE:',I2,5X/)
J=NUMBER OF THE CYCLE IN EACH UNCONSTRAINED MINIMIZATION
----------------------------------------------------------------
CALL GRAD(DES,DG,G)
CALL HESSI(DES,DG,HES,G,S)
CALL QUAD(DES,S,N,OPTP,FALFA)
DO 20 I=1,N
DES(I)=DES(I)+OPTP*S(I)
IF(NTIM.EQ.1) PHIOLD=FALFA
IF(NTIM.EQ.1) GO TO 10
PHINEW=FALFA
CHECK FOR CONVERGENCE USING PHI VALUE
------------------------------------------------
IF(((PHIOLD-PHINEW)/PHIOLD).LT.0.005) GO TO 100
PHIOLD=PHINEW
IF(NTIM.GT.7) GO TO 100
GO TO 10
RETURN
END
```

```
TO FIND THE MINIMUM IN ONE DIMENSIONAL SEARCH USING
QUADRATIC INTERPOLATION TECHNIQUE
SUBROUTINE QUAD(DES,S,N,ALFA,FALFA)
DIMENSION DES(3),S(3),DES1(3),DAL(3)
COMMON/QUAD1/STEP,EPS1,EPS2,MAXFIT,STLMT
COMMON/AREA/DES1
COMMON/ALLOW1/DAL
COMMON NFEVA
NFEVA=0
IFIT=0
A=0.0
ST=STEP
NORMALISATION OF S(I)
S1=0.0
DO 5 I=1,N
S1=S1+S(I)*S(I)
CONTINUE
S11=SQRT(S1)
DO 7 J=1,N
S(J)=S(J)/S11
CONTINUE
IK=1
FA=ONEDFN(DES,N,S,A)
F1=ONEDFN(DES,N,S,ST)
IF(F1.LT.FA) GO TO 20
FC=F1
C=ST
IF(ST.LT.STLMT) GO TO 120
ST=ST/2.0
B=ST
F1=ONEDFN(DES,N,S,ST)
IF(F1.LT.FA) GO TO 50
GO TO 10
FB=F1
B=ST
ST=ST*2.0
IK=IK+1
F2=ONEDFN(DES,N,S,ST)
IF(F2.GT.FB) GO TO 40
IF(IK.GT.15) GO TO 35
A=B
FA=FB
FB=F2
B=ST
GO TO 30
FC=F2
C=ST
GO TO 60
FB=F1
GO TO 60
FITTING THE QUADRATIC
C1=FA*(B-C)
C2=FB*(C-A)
C3=FC*(A-B)

C4=C1*(B+C)
C5=C2*(C+A)
C6=C3*(A+B)
ALFA=0.5*(C4+C5+C6)/(C1+C2+C3)
FALFA=ONEDFN(DES,N,S,ALFA)
```

```
DET=(A-B)*(B-C)*(C-A)
P=-(C1*B*C+C2*C*A+C3*A*B)/DET
Q=(C4+C5+C6)/DET
R=-(C1+C2+C3)/DET
HALFA=P+Q*ALFA+R*ALFA**2
DFALFA=Q+2.0*R*ALFA
ERR=ABS(((HALFA)-FALFA)/FALFA)
IFIT=IFIT+1
TYPE 70,A,FA,B,FB,C,FC,ALFA,FALFA,HALFA,ERR
FORMAT(5X,10E12.4)
IF(IFIT.GT.MAXFIT) GO TO 140
IF((ERR.LE.EPS1).AND.(DFALFA.LT.EPS2)) GO TO 140
IF(B.LT.ALFA) GO TO 80
IF(FB.GT.FALFA) GO TO 100
A=ALFA
FA=FALFA
GO TO 60
IF(FB.GT.FALFA) GO TO 90
C=ALFA
FC=FALFA
GO TO 60
A=B
FA=FB
B=ALFA
FB=FALFA
GO TO 60
C=B
FC=FB
B=ALFA
FB=FALFA
GO TO 60
TYPE  130
FORMAT(/,10X,'FUNCTION DOES NOT DECREASE ALONG THIS DIRECTION')
ALFA=0.0
FALFA=F1
GO TO 150
ALFAS=ALFA
TYPE  160,ALFA
FORMAT(/,10X,'MINIMUM IS AT ALFASTAR=',F10.4)
TYPE  170,IFIT,NFEVA
FORMAT(/,10X,'NUMBER OF QUADRATIC FITS:',I2,/10X,'TOTAL NO. OF
1 EVALUATIONS :',I3/)
GO TO 500
TYPE 400
FORMAT(/10X,'CAN NOT TRAP IN FIFTEEN JUMPS'/)
RETURN
END
```

```
SUBROUTINE FOR GRADIENT DEVELOPMENT
SUBROUTINE GRAD(DES,DG,GG)
)IMENSION SEC(10),DG(10,3),DGB(10),DGF(10)
)IMENSION FAM(3),GG(3)
)IMENSION CONS(10),DES(3)
COMMON/CONS1/NCON,NVAR
COMMON/STAGE/NSTAGE,NTIM,NSQ
COMMON/METHOD/NTYPN,NTYPL,NTYPE
COMMON/AREA9/CONS
COMMON/ET/AB
COMMON/PARAM/PT,COSRA,AM,COC
COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
)O 5 I=1,NCON
)GF(I)=0.0
)GB(I)=0.0
CONTINUE
)O 30 J=1,NVAR
IF(DES(J).EQ.0.0) GO TO 6
IF(J.EQ.NVAR) HJ=0.1*DES(J)
IF(J.EQ.NVAR) GO TO 7
HJ=0.01*DES(J)
GO TO 7
HJ=0.01
DES(J)=DES(J)+HJ
CALL ANALY(DES)
DO 10 I=1,NCON
DGF(I)=CONS(I)
CONTINUE
DES(J)=DES(J)-2.0*HJ
CALL ANALY(DES)
)O 15 I1=1,NCON
)GB(I1)=CONS(I1)
CONTINUE
DES(J)=DES(J)+HJ
)O 20 I=1,NCON
DG(I,J)=(DGF(I)-DGB(I))/(2.0*HJ)
CONTINUE
CONTINUE
CALL ANALY(DES)
)O 70 I=1,NVAR
SUM=0.0
)O 60 J=1,NCON
IF(NTYPE.EQ.3) GO TO 57
IF(NTYPE.EQ.2) GO TO 53
IF(CONS(J).GT.TRAP) GO TO 55
SUM=SUM-DG(J,I)/(TRAP**2)
GO TO 60
IF(CONS(J).GT.TRAP) GO TO 55
SUM=SUM+(2.0*CONS(J)/TRAP-3.0)*DG(J,I)/TRAP**2
```

```
      GO TO 60
57    IF(CONS(J).GT.TRAP) GO TO 55
      UT=CONS(J)/TRAP-1
      SUM=SUM+(3.0*UT**2+2.0*UT-1.0)*DG(J,I)/TRAP**2
      GO TO 60
55    SUM=SUM-DG(J,I)/(CONS(J)**2)
60    CONTINUE
      SUM=SUM*R1
      SEC(I)=SUM
70    CONTINUE
      DES3=DES(3)
      FAM(1)=(PI*DES(2)*COC*(1.0+COSRA*DES3))
      FAM(2)=(PI*DES(1)*COC*(1.0+COSRA*DES3))
      FAM(3)=PI*DES(2)*DES(1)*COSRA*COC
      DO 80 I2=1,NVAP
      GG(I2)=FAM(I2)+SEC(I2)
80    CONTINUE
      RETURN
      END
```

```
SUBROUTINE FOR HESSIAN MATRIX DEVELOPMENT
SUBROUTINE HESSI(DES,DG,HES,G,S)
DIMENSION HES(3,3),SAM(3,3),DG(10,3),G(3),A(3)
DIMENSION CONS(10),DES(3),D(3),AL(3),AMG(3),T(3),S(3)
COMMON/CONS1/NCON,NVAR
COMMON/METHOD/NTYPN,NTYPL,NTYPE
COMMON/AREA9/CONS
COMMON/ET/AH
COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
COMMON/PARAM/PI,COSRA,AM,COC
DO 5 I=1,NVAR
DO 5 J=1,NVAR
SAM(I,J)=0.0
HES(I,J)=0.0
CONTINUE
DO 10 J=1,NVAR
DO 10 K=1,J
SUM=0.0
DO 20 I=1,NCON
IF (NTYPE.EQ.2) GO TO 14
IF(NTYPE.EQ.3) GO TO 15
IF(CONS(I).LE.TRAP) GO TO 16
SUM=SUM+2.0/(CONS(I)**3)*(DG(I,J)*DG(I,K))
GO TO 20
IF(CONS(I).LE.TRAP) GO TO 17
SUM=SUM+2.0*DG(I,J)*DG(I,K)/(CONS(I)**3)
GO TO 20
SUM=SUM+DG(I,J)*DG(I,K)*(6.0*(CONS(I)/TRAP-1.0)
1+2.0)/(TRAP**3)
GO TO 20
SUM=SUM+2.0*DG(I,K)*DG(I,J)/(TRAP**3)
CONTINUE
SUM=SUM*R1
HES(J,K)=SUM
HES(K,J)=HES(J,K)
CONTINUE
DES3=DES(3)
SAM(1,2)=PI*COSRA*DES3*COC
SAM(1,3)=PI*COSRA*DES(2)*COC
SAM(2,3)=PI*COSRA*DES(1)*COC
SAM(2,1)=SAM(1,2)
SAM(3,1)=SAM(1,3)
SAM(3,2)=SAM(2,3)
DO 30 I=1,NVAR
DO 30 J=1,I
HES(I,J)=HES(I,J)+SAM(I,J)
HES(J,I)=HES(I,J)
CONTINUE
TYPE 50,((HES(I,J),J=1,NVAR),I=1,NVAR)
FORMAT(/10X,'HESSIAN MATRIX:',/3(30X,3(E14.8,5X)/)/)
TRANSFORMING HESSIAN MATRIX IN LDLT FORM
DO 80 I=1,NVAR
AMG(I)=0.0
AL(I)=0.0
D(I)=0.0
CONTINUE
D(1)=HES(1,1)
AMG(1)=HES(1,2)
AL(1)=AMG(1)/D(1)
D(2)=HES(2,2)-AL(1)*AMG(1)
AMG(2)=HES(1,3)
```

```
AMG(3)=HES(2,3)-AL(1)*AMG(2)
AL(2)=AMG(2)/D(1)
AL(3)=AMG(3)/D(2)
D(3)=HES(3,3)-AL(2)*AMG(2)-AL(3)*AMG(3)
DO 200 I=1,NVAR
 A(I)=0.0
T(I)=0.0
CONTINUE
DO 210 J=1,NVAR
IF(D(J).LT.0.0) GO TO 205
CONTINUE
CALL MATINV(HES,NVAR,G,S)
TYPE 206,(S(I),I=1,3)
FORMAT(/,5X,'MATINV',3(5X,F14.4)/)
RETURN
SOLVING LTT=A
DO 207 I=1,NVAR
IF(D(I).LE.0.0) A(I)=1.0
CONTINUE
T(3)=A(3)
T(2)=A(2)-AL(3)*T(3)
T(1)=A(1)-AL(1)*T(2)-AL(2)*T(3)
SUM1=0.0
DO 220 J=1,NVAR
SUM1=SUM1+T(J)*G(J)
CONTINUE
IF(SUM1.GT.0.0) GO TO 240
GO TO 230
DO 235 I=1,NVAR
S(I)=T(I)
CONTINUE
TYPE 500,(S(I),I=1,3)
FORMAT(/5X,'MOD SEARCH DIRE:',3(5X,E14.8)/)
GO TO 250
DO 245 I=1,NVAR
S(I)=-T(I)
CONTINUE
TYPE 300,(S(I),I=1,3)
FORMAT(/,5X,3(5X,F12.4)/)
RETURN
END
```

```
FUNCTION ONEDFN(DES,N,S,A)
DIMENSION DES(3),DES1(3),S(3)
COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
COMMON/AREA9/CONS
COMMON/AREA/DES1
COMMON NFEVA
DO 10 I=1,N
DES1(I)=DES(I)+A*S(I)
TYPE*,DES1
TYPE*,A
DO 15 I=1,N
IF(DES1(I).GT.0.0) GO TO 15
GO TO 17
CONTINUE
GO TO 18
CALL ANEG(DES1,S,EK)
A=EK
DO 19 J=1,N
DES1(J)=DES(J)+EK*S(J)
CONTINUE
CALL ANALY(DES1)
CALL PENALT(DES1,FCONS)
ONEDFN=OBJFN(DES1,N)+FCONS
TYPE*,ONEDFN
RETURN
END

FUNCTION DFUN(DES,N,S,A)
DIMENSION DES(3),S(3),G(3),DES1(3)
COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
COMMON NFEVA
DO 3 I=1,N
DES1(I)=DES(I)+S(I)*A
CONTINUE
CALL GRAD(DES1,DG,G)
F=0.0
DO 5 I=1,N
F=F+S(I)*G(I)
DFUN=F
RETURN
END

FUNCTION OBJFN(DES,N)
DIMENSION DES(3)
COMMON NFEVA
COMMON/PARAM/PI,COSRA,AM,COC
COMMON/ET/AH
DES3=DES(3)
OBJFN=(PI*DES(1)*DES(2)*(1.0+COSRA*DES3))*COC
NFEVA=NFEVA+1
RETURN; END
```

```
**********************************************************************
ANALYSIS OF CHIMNEY
SUBROUTINE ANALY(DFS)
FCU=UNIT WEIGHT OF CONCRETE
DT=MEAN DIAMETER OF CHIMNEY AT TOP
DTO=OUTER DIAMETER OF CHIMNEY AT TOP
DTI=INNER DIAMETER OF CHIMNEY AT TOP
DLO=OUTER DIAMETER OF LINING AT TOP
DBLI=INNER DIAMETER OF LINING AT SECTION UNDER CONSIDERATIO
DBLO=OUTER DIA OF LINING AT THE SECTION UNDER CONSIDERATION
DLI=INNER DIA OF LINING AT TOP
WCON=WEIGHT OF CONCRETE SHELL
WLIN=WEIGHT OF LINING
SP=SHAPE FACTOR
DB=MEAN DIAMETER OF CHIMNEY AT BOTTOM(SECTION UNDER CONSIDERATIO
DBO=OUTER DIA CHIMNEY SHELL AT THE ABOVE SECTION
DBI=INNER DIA OF CHIMNEY SHELL AT THE ABOVE SECTION
DS=MEAN DIAMETER OF AIR SPACE IN MET
**********************************************************************
DIMENSION DES(3),CONS(10)
COMMON/AREA9/CONS
COMMON/AREA4/W,AMOM,ALP,WD,WC,WCD
COMMON/AREA7/ODBO(11),OTH(11),SABYG(30)
COMMON/AREA2/TOTL,TLIN,AIRGAP
COMMON/AREA5/P,PM,BETA,EBYR
COMMON/STAGE/NSTAGE,NTIM,NSU
COMMON/ET/AH
COMMON/PARAM/PI,COSRA,AM,COC
DATA [illegible]IB,THIT,DBASE,HEIGHT/0.225,0.16,8.875,80.0/
DATA AIRGAP,TLIN,FCU,FCL/0.1,0.115,2400.0,2100.0/
DATA SP,SGP,TMAX,TOUT,RO/0.10,0.115,160.0,4.0,0.5/
DATA CC,C[illegible],AK1,AK2,AKS/1.488,1.25,50.0,58.59,10.2/
DATA [illegible],ES,CL[illegible]/161818.18,2E6,11E-6/
DATA [illegible],[illegible],CF/0.02,[illegible],1.0/
DATA FC,FSY/250.0,2600/
DATA C[illegible],C5,A[illegible]/160.0,80.0,11.0/
[illegible]=[illegible].7
[illegible]=1.1
[illegible]=[illegible].5
[illegible]=0.115
IF([illegible].GE.1) GO TO 10
IF(NSTAGE.EQ.1) TOTL=HEIGHT
[illegible]=TOTL-AH
TYPE 5,TOTL
FORMAT(/10X,'THE PRESENT SECTION IS AT A DISTANCE OF ',F4.1,
11X,'METRES FROM THE BOTTOM'/)
AINC=FLOAT(NSTAGE)
STARTING DESIGN VARIABLES
-------------------------
DES(1)=DT+AINC*0.4
IF(NSTAGE.LE.5) DES(2)=0.16
IF((NSTAGE.GE.6).AND.(NSTAGE.LE.7)) DES(2)=0.20
IF(NSTAGE.GE.8) DES(2)=0.275
DES(3)=0.005
MODIFIED ANALYSIS PROGRAM
```

```
ETTU=HEIGHT-TOTL
IF(NSTAGE.NE.1) GO TO 1000
THI=0.15
DI=4.85
GO TO 1500
DI=DDBO(NSTAGE-1)
THI=DTH(NSTAGE-1)
DTO=DT+THI
DTI=DT-THI
DLO=DTI-2.0*AIRGAP
DLI=DLO-2.0*TLIN
DB=DES(1)
THI=DES(2)
DBO=DB+THI
DBI=DB-THI
DBLO=DBI-2.0*AIRGAP
DBL=DBLO-TLIN
DS=DBLO+AIRGAP
DBLI=DBLO-2.0*TLIN
AMASCT=DBO*DBO+DTO*DTO+DBO*DTO-DBI*DBI-DTI*DTI-DBI*DTI
WCONT=AMASCT*FCU*AH*PI/12.0
AMASLT=DBLO*DBLO+DLO*DLO+DBLO*DLO-DBLI*DBLI-DLI*DLI-DBLI*DLI
WLINT=AMASLT*AH*FCL*PI/12.0
W=WO+WCONT+WLINT
WC=WCO+WCONT
IF(NSTAGE.GT.6) GO TO 42
ANUM1=AMASLT+DBLO*DLO-DBLI*DLI+2.0*(DLO*DLO-DLI*DLI)
HLBART=ANUM1*AH*0.25/AMASLT
ANUM=AMASCT+2.0*(DTO*DTO-DTI*DTI)+(DBO*DTO-DBI*DTI)
HCBART=ANUM/AMASCT*0.25*AH
CG FOR BOTTOM SECTION
DB=DES(1)
[illegible]=DES(2)
[illegible]=[illegible].275
[illegible]=[illegible].275
[illegible]=[illegible]+[illegible]
[illegible]=[illegible]-[illegible]
[illegible]=[illegible]-2.0*AIRGAP
[illegible]=[illegible]-2.0*[illegible]
[illegible]=[illegible]**2+[illegible]**2-DBBI**2-DBI**2+DBBO*DBO-DBBI*DBI
[illegible]=[illegible]**2+[illegible]**2+DBBLO*DBLO-DBBLI*DBLI-DBBLI**2-DBLI**2
[illegible]=[illegible]*TOTL*FCU*PI/12.0
[illegible]=[illegible]*PI*TOTL*FCL/12.0
[illegible]=[illegible]+2.0*([illegible]**2-DBI**2)+DBO*DBBO-DBI*DBBI
[illegible]2=[illegible]+2.0*(DBLO**2-DBLI**2)+DBBLO*DBLO-DBBLI*DBLI
HCBARB=TOTL*A[illegible]1*0.25/AMASCB
HLBARB=TOTL*ANUM2*0.25/AMASLB
ANU=WCONT*(HCBART+TOTL)+WLINT*(HLBART+TOTL)
ANU=ANU+WCONB*HCBARB+WLINB*HLBARB+WO*0.5*(HEIGHT+TOTL)
DNM=WCONT+WLINT+WLINB+WCONB+WO
HBAR=ANU/DNM
WTOT=WCONB+WLINB+W
TH=HEIGHT
TP=48.0*SQRT(WTOT*5.84373E-10)
TP=TP*10
I=IFIX(TP)
```

```
IF(I.LE.3.0) GO TO 5010
SAB=SABYG(I)-(TP-I)*(SABYG(I)-SABYG(I+1))
GO TO 5020
ALPAH=0.056
GO TO 5030
ALPAH=SAB*0.2
ALN=HEIGHT-TOTL
TP=TP/10.
AMOM=ALPAH*WTOT*HBAR*(0.6*(ALN/TH)**0.5+0.4*(ALN/TH)**4.0)
GO TO 200
IF((TOTL.LE.56.0).AND.(TOTL.GE.24.0)) GO TO 90
IF((TOTL.LT.24.0).AND.(TOTL.GE.0.0)) GO TO 110
TYPE 45
FORMAT(10X,'ERROR IN CALCULATING HEIGHT'/)
ERROR=1.0
GO TO 300
PRE1=122.0
H1=24.0
PRE2=115.0          TOTL
H2=ETTO-24.0
WPRE1=0.5*PRE1*(5.0+ODBO(3)+OTH(3))*SF*H1
WPRE2=0.5*PRE2*(ODBO(3)+OTH(3)+DBO)*SF*H2
AMOM1=WPRE1*(H1/3.0*((10.0+ODBO(3)+OTH(3))/(5.0+ODBO(3)+OTH(3)))
1+H2)
AMOM2=WPRE2*(H2/3.0*(2.0*(ODBO(3)+OTH(3))+DBO)/(DBO+ODBO(3)+
1OTH(3)))
AMOM=AMOM1+AMOM2
GO TO 200
FORMAT(/10X,'WEIGHT OF CHIMNEY ABOVE THE PRESENT SECTION',F12.2,
11X,'KG',/10X,'MOMENT AT THAT SECTION IS',F12.2,1X,'KG-MET'/)
H1=24.0
H2=32.0
H3=ETTO-56.0
PRE1=122.0
PRE2=115.
PRE3=1[illegible].0
WPRE1=0.5*PRE1*(5.0+ODBO(3)+OTH(3))*H1*SF
WPRE2=0.5*PRE2*(ODBO(3)+ODBO(7)+OTH(7)+OTH(3))*H2*SF
WPRE3=0.5*PRE3*(ODBO(7)+OTH(7)+DBO)*H3*SF
ALEV1=0.3333*H1*(10.0+ODBO(3)+OTH(3))/(5.0+ODBO(3)+OTH(3))+H2+H3
ALEV2=0.3333*(2.0*(ODBO(3)+OTH(3))+ODBO(7)+OTH(7))
ALEV2=ALEV2/(ODBO(3)+OTH(3)+ODBO(7)+OTH(7))*H2+H3
ALEV3=0.3333*(2.0*(ODBO(7)+OTH(7))+DBO)/(ODBO(7)+OTH(7)+DBO)*H3
AMOM1=WPRE1*ALEV1
AMOM2=WPRE2*ALEV2
AMOM3=WPRE3*ALEV3
AMOM=AMOM1+AMOM2+AMOM3
GO TO 200
ECCEN=AMOM/W
EBYR=ECCEN*2.0/DES(1)
IF(NSTAGE.EQ.9) GO TO 260
IF(EBYR.LT.0.5) GO TO 300
CALL ANEUT(DES)
GO TO 300
BETA=3.0/DES(1)
EXPR=((PI-BETA)**2-(SIN(BETA))**2)/((PI-BETA)*COS(BETA)
```

```
1*SIN(BETA))
EXPR=(EXPR-3.0*SIN(BETA))/(2.0*(PI-BETA))
IF(EBYR.LE.EXPR) GO TO 300
CALL ANEUT(DES)
TYPE 250,EBYR,BETA,P,ALP
FORMAT(/10X,'EBYR=',F5.2,10X,'BETA=',F5.2,10X,'P=',F6.4,10X,
1'ALPA=',F10.4/)
-----------------------------------------------------------------

FORMULATION OF CONSTRAINTS FOR CHIMNEY DESIGN
CLIN=COEFFICIENT OF LINIEAR EXPANSION FOR CONCRETE
TLIN=THICKNESS OF LINING
STRESSE DUE TO WIND AND DEAD LOAD ALONE
STRESSES DUE TO EARTHQUAKE AND DEAD LOAD
INSERT FOR EARTHQUAKE
------------------------------------------------------------
IF(NSTAGE.EQ.9) GO TO 350
IF (EBYR.LT.0.5) GO TO 500
GO TO 400
IF(EBYR.LE.EXPR) GO TO 550
GO TO 400
AM=2.0*(EBYR+SIN(BETA)/(PI-BETA))
AM=AM*((PI-BETA)*COS(BETA)+SIN(BETA))
AM=AM/((PI-BETA)-0.5*SIN(2.*BETA)-2.0*(SIN(BETA))**2/(PI-BETA))
SCVD=W/((PI-BETA)*DES(1)*DES(2)*1.0E4)*AM
SCVDW=SCVD/(1.0+DES(2)/DES(1))
SSVDW=SCVDW*AM
GO TO 555
DIF=COS(BETA)-COS(ALP)
DES3=DES(3)
DENOM1=(1.0-DES3)*(SIN(ALP)-ALP*COS(ALP))-(1.0-DES3+AM*
1DES3)*(SIN(BETA)-BETA*COS(ALP))-AM*DES3*PI*COS(ALP)
SCVD = [illegible]/(DES(1)*DES(2)*1.0E4)*(DIF/DENOM1)
[illegible] = [illegible]*SCVD*(1.0+COS(ALP))/DIF
[illegible]=[illegible]S(2)/(DES(1)*COS(BETA)*DIF)

[illegible] GE.[illegible]) GO TO 570
[illegible]
[illegible] = [illegible]*(1.0+2.0*EBYR)/(DES(1)*DES(2)*PI*1.0E4)
[illegible] =[illegible]/(1.0+(DES(2)/DES(1)))
[illegible] = [illegible]*SC[illegible]
[illegible] GO TO 570
[illegible]( )=1.[illegible]-SC[illegible]/(0.3[illegible]*FC)
[illegible]( )=1.[illegible]-SS[illegible]/(0.57*FSY)

[illegible]( )=1.[illegible]-SC[illegible]*(1.[illegible])/(0.35*FC)
[illegible]( )=1.[illegible]-SSV[illegible]/(0.57*FSY)
[illegible]
[illegible](1)=1.[illegible]-SCV[illegible]*(1.[illegible])/(0.4*FC)
[illegible](2)=1.[illegible]-SSV[illegible]/(0.5*FSY)
[illegible] TEMPERATURE ONLY
------------------------------------
[illegible] = .115
[illegible]=[illegible](2)*[illegible]/(CC*DES(1))
[illegible]=1.[illegible]/([illegible]*[illegible])+(TLIN*DBLI)/(RQ*CB*DBL)+DBLI/(AKS*DS)+
```

```
2(DES(2)*DBL1)/(CC*DES(1))+DBLI/(AK2*DBD)
TX=COF*(TMAX-TOUT)/DEN2
DES3=DES(3)
AZ=(DES(2)-0.0635)/DES(2)
AK=-DES3*AM+SQRT(DES3*AM*(DES3*AM+2.0*AZ))
SSVT=CLIN*(AZ-AK)*TX*ES
SCVT=CLIN*AK*TX*EC
STRESSES DUE TO W.L,D.L. AND TEMPERATURE
---------------------------------------------
AKCOMB=-DES3*AM+(DES3*AM*(DES3*AM+2.0*AZ)+2.0*AK*(1.0+
1DES3*AM)*SCVDW/SCVT)**0.5
IF (AKCOMB.LE.1.0) GO TO 5000
SCVDWT=SCVDW+SCVT/AK*((2.0*AM*DES3*AZ+1.0)/(2.0*(1.0+AM*DES3
1)))
GO TO 5200
SCVDWT=SCVT*AKCOMB/AK
CONS(3)=1.0-SCVDWT/(0.50*FC)
REL=AM*AZ*EC*CLIN*TX
IF(SSVDW.GT.REL) GO TO 5255
SSVDWT=SSVT/(AZ-AK)*(AZ+AM*DES3-SQRT(AM*DES3*(AM*DES3+2.0*
1AZ)-2.0*AM*DES3*(AZ-AK)*SSVDW/SSVT))
CONS(4)=1.0-SSVDWT/(0.65*FSY)
GO TO 5266
CONS(4)=1.0
STRESSES DUE TO DEAD LOAD AND TEMPERATURE
---------------------------------------------
SSVD=MAX VERTICAL STRESS(COMP) IN STEEL DUE TO D.L ONLY
SCVD=MAX VERT. STRESS(COMP) IN CONCRETE DUE TO D.L ONLY
IF(NSTAGE.EQ.9) GO TO 585
SCVD=W/(PI*DES(1)*DES(2)*1.0E4)
GO TO 587
SCVD=[illegible]/((PI-BETA)*DES(1)*DES(2)*1.0E4)
SCV[illegible]=SCVD/(1.0+[illegible]ES(2)/DES(1))
[illegible]=A[illegible]*SCVD
[illegible] STRESS IN COMP IN CONCRETE ON LEEWARD SIDE
---------------------------------------------
[illegible]=-[illegible]*[illegible]ES3+SQRT(AM*DES3*(AM*DES3+2.0*AZ)+2.0*AK*(1.0+AM*
[illegible])*SC[illegible]/SCVT)
[illegible].1.[illegible]) GO TO 3000
[illegible]=[illegible]VT*[illegible])/AK
[illegible] 310
SC[illegible]=SCV[illegible]+SCVT/AK*(2.0*AM*DES3*AZ+1.0)/(2.0*(1.+AM*DES3))
S[illegible]=SSVT/(AZ-AK)*(AZ+AM*DES3-(AM*DES3*(DES3*AM+2.0*AZ)-
12.[illegible]*[illegible]*[illegible]ES3*(AZ-AK)*SSVD/SSVT)**0.5)
TOTAL STRESS IN STEEL ON WIND WARD SIDE
---------------------------------------------
CHECK HERE
CONS(5)=1.0-SCVDT/(0.33*FC)
CONS(6)=1.0-SSVDT/(0.55*FSY)
CONS(7)=333.333*DES3-1.0
CONS(8)=1.0-20.0*DES3
IF(NSTAGE.EQ.1) GO TO 600
JJ=NSTAGE-1
CONS(9)=1.0+(DES(2)-DDBO(JJ)-DTH(JJ))/DES(1)
GO TO 625
CONS(9)=1.0+(DES(2)-5.0)/DES(1)
```

```
[illegible] IF(DES(1).GT.6.0) GO TO 650
      CONS(10)=DES(2)/0.15-1.0
      GO TO 700
650   CONS(10)=DES(2)/0.15-(DES(1)-6.0)/180.-1
700   RETURN
      END
```

```
AVOIDING NEGATIVE DESIGN VARIABLES IN THE DESIG
SUBROUTINE ANEG(DES,S,EK)
DIMENSION DES(3),S(3),DAL(3),DES1(3)
DIMENSION AJSTAR(10),JSET(10),ALPE(10)
COMMON/CONS1/NCON,NVAR
COMMON/ALLOW1/DAL
COMMON/CONS2/C1,C2,R1,SIGMA,TRAP
DO 3 I=1,10
AJSTAR(I)=0.0
JSET(I)=0
CONTINUE
M=0
DO 10 I=1,NVAR
IF(DES(I).GT.0.) GO TO 10
M=M+1
AJSTAR(M)=DES(I)
JSET(M)=I
CONTINUE
DO 20 K=1,M
J=JSET(K)
ALPE(K)=(DAL(J)*(1.0+TRAP)-AJSTAR(J))/S(J)
CONTINUE
IF(M.EQ.1) SMALL=ALPE(M)
IF(M.EQ.1) GO TO 100
FINDING THE SMALLEST ALPE(J)
SMALL=ALPE(1)
DO 30 K1=2,M
IF(SMALL.LT.ALPE(K1)) GO TO 30
SMALL=ALPE(K1)
CONTINUE
EK=SMALL
RETURN
END
```

```
SUBROUTINE ANEUT(DES)
DIMENSION DES(3)
COMMON/AREA5/P,PM,BETA,EBYR
COMMON/STAGE/NSTAGE,NITM,NSQ
COMMON/DIA/DBO,DBI,DB,DBL,DBLI,DBLO,DS
COMMON/AREA4/WTOT,AMOM,ALPA3
COMMON/PARAM/PI,COSRA,AM
PI=3.1415926
P=DES(3)
PM=AM*DES(3)
IF (NSTAGE.EQ.9)GO TO 5
BETA=0.0
GO TO 6
BETA=3.0/DES(1)
ALPA1=0.8
GO TO 7
ALPA1=0.1
ALPA2=2.5
ALPA3=(ALPA1*OG(ALPA2)-ALPA2*OG(ALPA1))/(OG(ALPA2)-OG(ALPA1))
IF(OG(ALPA3).LT.0.001) GO TO 30
IF((OG(ALPA3)*OG(ALPA1)).LT.0.0) GO TO 20
ALPA1=ALPA3
GO TO 10
ALPA2=ALPA3
GO TO 10
RETURN
END

FUNCTION  OG(ALPA)
COMMON/AREA5/P,PM,BETA,EBYR
COMMON/PARAM/PI,COSRA,AM
A=0.5*(1.0-P)*(ALPA-SIN(ALPA)*COS(ALPA))-0.5*(1.0-P+PM)*(BETA+
1SIN(BETA)*COS(BETA)-2.0*COS(ALPA)*SIN(BETA))+0.5*PM*PI
B=(1.0-P)*(SIN(ALPA)-ALPA*COS(ALPA))-(1.0-P+PM)*
1(SIN(BETA)-BETA*COS(ALPA))-PM*PI*COS(ALPA)
OG=A-B*EBYR
RETURN
END
```